Signal processing for high resolution pulse width modulation based digital-to-analogue conversion.

Goldberg, Jason M

Signal Processing for High Resolution Pulse Width Modulation Based Digital-To-Analogue Conversion

A Thesis Submitted to
The University of London
in Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

By
Jason M. Goldberg
November 1992

CONTAINS DISKETTE

UNABLE TO COPY

CONTACT UNIVERSITY

IF YOU WISH TO SEE

THIS MATERIAL

Abstract

This thesis investigates techniques for "digital power amplification" or the transforming of a digital signal directly into analogue power with no intermediate low power digital-to-analogue converter (DAC) stage. As opposed to their analogue counterparts, it is possible for digital amplifiers to be compact, efficient, and highly linear.

In particular, we consider the suitability of digital Pulse Width Modulation (PWM) techniques for class D digital power amplification. Earlier work has shown that while offering certain advantages, 16 bit quality PWM was not practical to implement due to excessive modulator clock speed. Also, nonlinearities inherent to the modulation processes typically used often gave rise to high levels of distortion.

We specifically focus on the application of DSP techniques intended to both facilitate hardware implementation and increase the performance of PWM based amplifiers.

Oversampled Noise Shaping (ONS) is shown to make digital amplifiers more practical to realize in hardware. ONS networks are used to reduce the pulse width modulator's input signal wordlength (with negligible loss in baseband signal quality). This reduction in wordlength allows a corresponding reduction in modulator clock speed which is large enough to make a practical implementation possible. Particular attention is paid to the special design considerations for ONS networks used in conjunction with PWM.

We also propose a new scheme to eliminate PWM distortion called "Pseudo Natural PWM" (PNPWM). It consists of a premodulation signal processing algorithm designed to eliminate the harmonic distortion generated by the uniform sampling PWM (UPWM) modulators typically proposed for use in digital amplifiers. The algorithm digitally approximates the sampling instants associated with the harmonic distortion free natural sampling PWM (NPWM) process. These sampling instants are applied to a conventional UPWM modulator to achieve near to distortion free NPWM performance (previously thought to be possible only in continuous time analogue systems).

For my parents

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Mark Sandler. His guidance and helpful suggestions have played a key role in the success of this research project.

My thanks are also due to several colleagues and members of staff who have been particularly cooperative. Specifically, I was pleased to collaborate with Rod Hiorns, Jie Yu, and Dr. Rainer Nawrocki. I have benefited from their patience, insight, and determination.

In addition, I am indebted to my parents, Irving and Janet Goldberg, and to the rest of my family. Their continuous support and encouragement has been essential for the completion of this thesis.

Finally, I wish to thank Professor Charles Turner and the Department of Electronic and Electrical Engineering for the financial assistance which made this work possible.

Table of Contents

List of Figures

List of Tables

Chapter One
Introduction

Most signals encountered in the natural world are *analogue* in that they vary continuously with time and amplitude. However, analogue signals are very sensitive to noise and are therefore difficult to process accurately. In contrast, discrete time, discrete amplitude, *digital* signals are inherently less sensitive to noise. Over the past two decades, dramatic advances in digital integrated circuit technology have enabled the efficient and accurate processing, storage, transmission, and analysis of digital signals. Digital Signal Processing (DSP) techniques are now applied in fields as diverse as telecommunications, speech and audio processing, radar and sonar, geophysics, and image processing.

Consider for instance a typical digital audio application as shown in Fig. 1.1. An analogue signal is converted to a digital format via an analogue-to-digital converter (ADC), after which it may be processed, stored, transmitted, etc. completely in the digital domain. Eventually, the digital representation of the signal is converted back to the analogue domain via a digital-to-analogue converter (DAC).

Currently, digital techniques have developed to the point where processing can be carried out digitally with much greater accuracy than possible in the actual ADC and DAC procedures themselves. As such, a great deal of research effort has recently been directed toward extending ADC and DAC accuracy beyond present limitations. This is particularly true for 16 bit linear PCM digital audio where dynamic range requirements are near to 100dB.

Conventional high resolution ADC techniques are usually based on successive approximation or dual slope methods [Bl78]. High resolutions DACs typically use resistor-ladder networks or current divider techniques [Va82a, Lo82]. In general, these approaches require very high accuracy circuit components and often some form of laser trimming or dynamic element matching is necessary [Va82b]. This increases the cost and the complexity of manufacturing such converters.

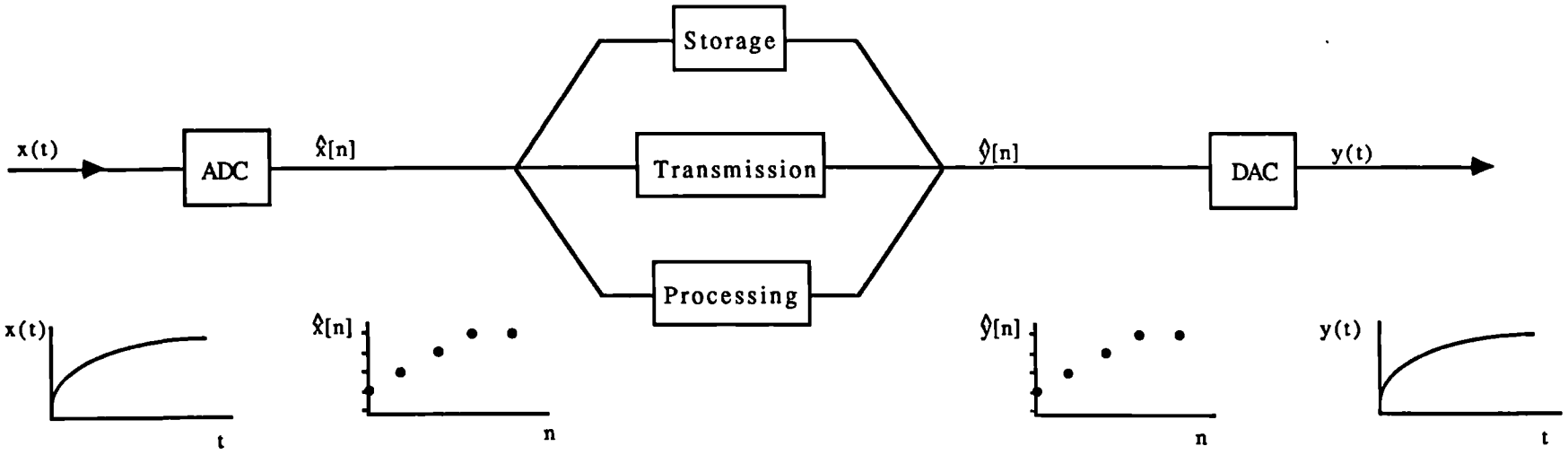


Fig. 1.1 General Digital Audio System

Recently, oversampled noise shaping (ONS) and sigma-delta modulation (SDM) have attracted much interest as alternatives to traditional conversion techniques. This is because they do not require precise trimming of circuit components, high precision sample and hold circuits, or analogue low pass filters with sharp transition bands. The relative lack of high accuracy components also make ONS and SDM particularly well suited to modern VLSI implementation methods.

Broadly speaking, ONS and SDM techniques achieve high levels of performance through a combination of oversampling, coarse quantization (typically one bit), error feedback, and digital filtering. Low order implementations have been used successfully in digital audio Compact Disc (CD) applications [Na87]. Unfortunately, higher order implementations are often susceptible to stability problems [Ca85]. However, multi-stage noise shaping (MASH) techniques have been found to avoid such problems by combining the effect of stable, low order noise shapers to achieve stable, high order performance [Ma87,Ma89]. These have also been used in high quality digital audio applications [Ph].

In spite of all this progress, the high quality output of today's CD players (or Digital Audio Tape (DAT) players) are still fed into conventional analogue amplifiers, which, in turn, drive loudspeakers. This is shown in Fig. 1.2. The typical analogue amplifier is known to be inefficient, nonlinear, and bulky. Even so, relatively little research on audio amplifiers has been directed toward the application of the types of digital techniques which have managed to revolutionize the earlier stages of audio processing. Currently the "digitization" of audio firmly ends at the output of the CD or DAT player.

This thesis explores techniques by which a digital signal can be converted *directly* into a corresponding high power analogue waveform (i.e., with no intermediate low power DAC stage). This is shown in Fig. 1.3. The low power DAC and the high power analogue amplifier are combined into a single power level DAC. Such a device can be thought of as a type of "digital power amplifier."

1.1 History and Motivation

In 1983 Sandler proposed the use of a class D Pulse Width Modulation (PWM) based digital power amplifier for digital audio applications [Sa83]. PWM is a pulse modulation technique whereby signals are conveyed as variations in the pulse widths of a high frequency pulse waveform. In *digital* PWM the pulse widths are quantized. High efficiency amplification of the two state PWM waveform was proposed to be achieved with fast MOSFET power switching circuits. This is shown in Fig. 1.4.

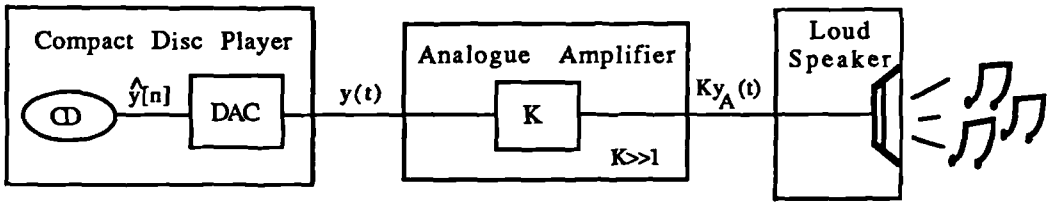


Fig. 1.2: Conventional Analogue Amplifcation

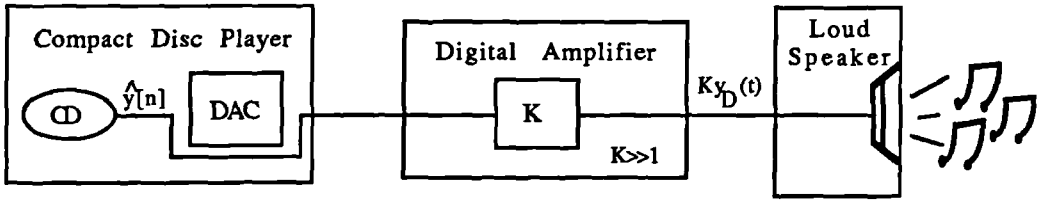
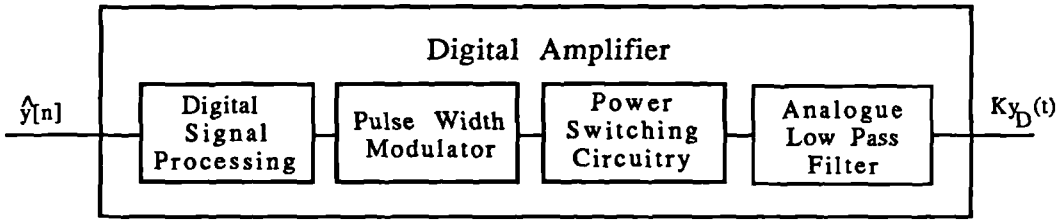
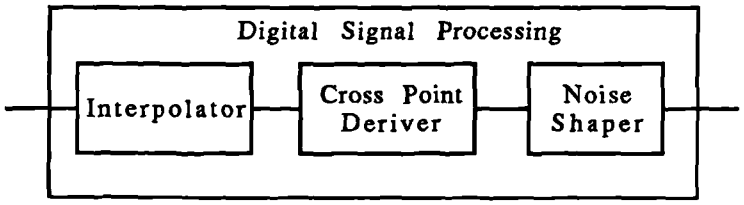


Fig. 1.3: Digital Amplifcation



(a)



(b)

Fig. 1.4: (a) PWM Based Digital Amplifier
(b) Expanded View of DSP Block

Aside from nonidealities in the power switching stage, two main difficulties continue to be associated with the PWM approach: *practicality* and *performance*. A 16 bit quality, 20kHz bandwidth implementation could not be realized easily in hardware due to the excessive modulator clock speeds required to resolve $2^{16} = 65536$ distinct pulse-widths in no less than $\frac{1}{44.1kHz} \approx 22.7\mu sec$ per pulse. Moreover, nonlinearities inherent to the specific modulation processes proposed for use in the amplifier made 16 bit performance very difficult to achieve.

This thesis will address the issues of practicality and performance. In particular, it is argued that the application of special DSP techniques prior to modulation (see Fig. 1.4) greatly enhances the potential for realizing a practical, linear 16 bit quality PWM based digital power amplifier.* We will be concentrating primarily on the signal processing techniques used to reduce modulator clock speed and to reduce (or eliminate) distortion. They include interpolation, noise shaping, and a special premodulation linearization algorithm. The specifics are outlined in the next section.

1.2 Structure of Thesis

Chapter Two introduces the various PWM modulation types considered in this thesis. PWM waveforms can be distinguished on the basis of how the samples modulating the waveform are obtained from the input signal. In so-called "uniform sampling PWM" (UPWM) the sampling instants of the samples that modulate the pulse waveform are uniformly spaced in time. Alternatively, in "natural sampling PWM" (NPWM) the sampling instants are nonuniformly spaced and, in fact, are signal dependent. Further classification is made with regard to whether a single edge or both edges of the PWM waveform are modulated. Additional distinctions are made for digital PWM (i.e., PWM with pulses that are quantized in width).

Where possible Fourier series expansions for the tone spectra associated with various modulation types are presented. Several PWM variants are compared on the basis of these

* At this stage it is worth noting that much of the work described in this thesis can also be viewed in the general context of ordinary, low power DACs. PWM techniques have recently been used as a means of avoiding circuit problems associated with the local one bit switched capacitor DACs often used in MASH type converters [Ma89]. Many of the results on linearizing PWM as well as on the interaction between the noise shaper and the pulse width modulator are relevant for such ONS/PWM based DACs. As such, unless otherwise indicated we will use the term "DAC" to refer to both conventional, low power DACs as well as high power DACs (i.e., digital power amplifiers).

spectra. It will be seen that all forms of PWM generate baseband distortion. This can be reduced by increasing the pulse repetition frequency of the PWM waveform. It is also stressed that NPWM systems do not exhibit any harmonic distortion. This important fact is used later in the thesis.

Chapter Three consists of a short review of the sample rate conversion techniques known as interpolation (sample rate increase) and decimation (sample rate decrease). These techniques play important roles in both the hardware implementation and software simulation of PWM based DACs. In particular, interpolation aids in the reduction of distortion while decimation is used as part of a narrow band spectral analysis procedure in the software simulation. After the basic approach is described, techniques for improving the computational efficiency of the sample rate change procedures are outlined.

Chapter Four is devoted to oversampled noise shaping (ONS) networks in general, as well as their specific use in PWM based DACs. As stated earlier, ONS techniques use a combination of oversampling, coarse quantization, and error feedback. Baseband SNR can be maintained in the oversampled, low wordlength output of the network at the expense of increased high frequency noise power.

An important problem for high quality digital PWM is the excessive speed at which the pulse width modulator must operate. A b bit modulator must be able to produce pulses of 2^b distinct widths within each pulse interval time. For a 16 bit, 20kHz audio system this implies master clock rates which are well into the GigaHertz range. We will see that ONS techniques can be exploited such that the modulator may operate at reasonable clock speeds. Later in the chapter, additional considerations in the design of ONS networks are presented. These are relevant in the tailoring of ONS networks for particular use in PWM based DACs. Specific designs of several ONS noise transfer functions (NTFs) are presented.

In Chapter Five, we address the issue of linearity in PWM based DACs. This is another important problem. In the past only UPWM was thought to be suitable for use in PWM conversion systems. This is because most digital input signals in some sense correspond to *uniformly* spaced samples of an analogue waveform. (Recall that in NPWM the sampling instants are nonuniformly spaced and signal dependent.) The use of UPWM modulators results in a harmonic distortion problem which cannot be eliminated without excessive pulse repetition frequencies. The techniques described in this chapter attempt to solve this problem by proposing the use of a discrete time, fully digital version of NPWM which is capable of offering distortion free performance at reasonable pulse repetition frequencies. We call this linearization approach "Pseudo-Natural PWM (PNPWM)." The basic idea is to approximate the width of the NPWM pulses associated with the underlying

analogue waveform (quantized samples of which form the actual digital input signal) and to apply these approximations (rather than the input signal) to an ordinary UPWM modulator. Approximation of the NPWM pulse widths is achieved with a DSP based signal approximation/root-finding procedure which we call "cross point derivation." Several algorithms of varying degrees of computational complexity and accuracy are presented.

Chapter Six describes the software which has been written to simulate the performance of the PWM based DACs considered in this thesis. The complicated interaction between the nonlinear stages comprising these DACs make the simulation a valuable tool for assessing overall system performance. The software is structured as a collection of small modules such that the function of each stage in the actual hardware implementation of a DAC is mimiced by one of these modules. This approach gives us the flexibility required to test a large variety of converters. There are additional design tools and programs which aid in the narrow band spectral analysis procedure used to evaluate the performance of the DACs.

Chapter Seven presents results from extensive computer simulations of many PWM based DACs. The chapter is divided into three main sections. The first concentrates on the simplest type of UPWM DAC (both with and without oversampling). Results are shown for a variety of sinusoidal inputs. Performance is examined as a function of pulse repetition frequency. Correspondence of simulation results with theoretically predicted performance is verified.

The next section considers ONS/UPWM DACs using the wide variety of ONS networks presented in Chapter Four. These systems are more realistic in that the modulators operate at clock speeds low enough to be realized in hardware. Under certain conditions, some newly discovered, undesirable baseband effects are shown to arise. The results indicate that *careful* selection of noise shaper NTF is necessary in order to avoid these effects.

The last section presents results from simulations of ONS/PNPWM DACs which use the pre-modulation linearization procedures described in Chapter Five. DAC performance is investigated for a variety of input signals. In all cases it is seen that the linearization procedures reduce UPWM harmonic distortion. In many instances the more accurate (but more computationally intensive) algorithms are capable of completely eliminating the distortion. The errors associated with each procedure are experimentally characterized in detail. These indicate which sources of error within each algorithm tend to dominate the overall error. The structure of the errors is also examined.

Finally, Chapter Eight summarizes the central ideas conveyed in each chapter and highlights the main results. Directions for future work are also indicated.

Chapter Two
Pulse Width Modulation

2.1 Introduction

The technique known as "Pulse Width Modulation" is part of the broad category of so called "pulse modulation" techniques. Communications systems employing such techniques arose as a direct consequence of the sampling theorem (i.e., of the fact that a continuous time signal of bandwidth less than f_bHz can be uniquely represented by uniformly spaced samples of the signal at a rate not less than $2f_b$ samples per second). Pulse modulation systems represent a message-bearing signal by using it to modulate some parameter of a periodic pulse train or "carrier waveform." It is possible to modulate the amplitude, position, or width of such pulses giving rise to Pulse Amplitude Modulation (PAM), Pulse Position Modulation (PPM), or Pulse Width Modulation (PWM), respectively. These three modulation types are shown in Fig. 2.1. In the past, pulse modulation schemes such as these became popular in communication systems because, among other things, they were particularly well suited for the time-multiplexing of several signals on a single transmission channel. The pulse time modulation (PTM) techniques, PPM and PWM, generally require a larger bandwidth but have an advantage over PAM in that the sample values of the modulating signal are not contained in the pulse *amplitudes* which are easily corrupted by channel noise. Hence there exists a tradeoff in PTM techniques between the bandwidth and signal-to-noise ratio (SNR) requirements. (See [Pa65] for more details.)

In our application, PTM techniques have additional important advantages. PTM based DACs can be constructed from digital counting circuits which may provide very accurate time base resolution. Such circuits in effect also guarantee the monotonicity of the DAC [Hi92a]. In addition, for digital power amplifiers, PTM systems have the advantage of being *two-level* modulation systems which can be implemented in a highly efficient on/off class D power switching stage. It is also easier to construct a class D power switching stage than a sufficiently accurate, stable power amplification stage with a large number

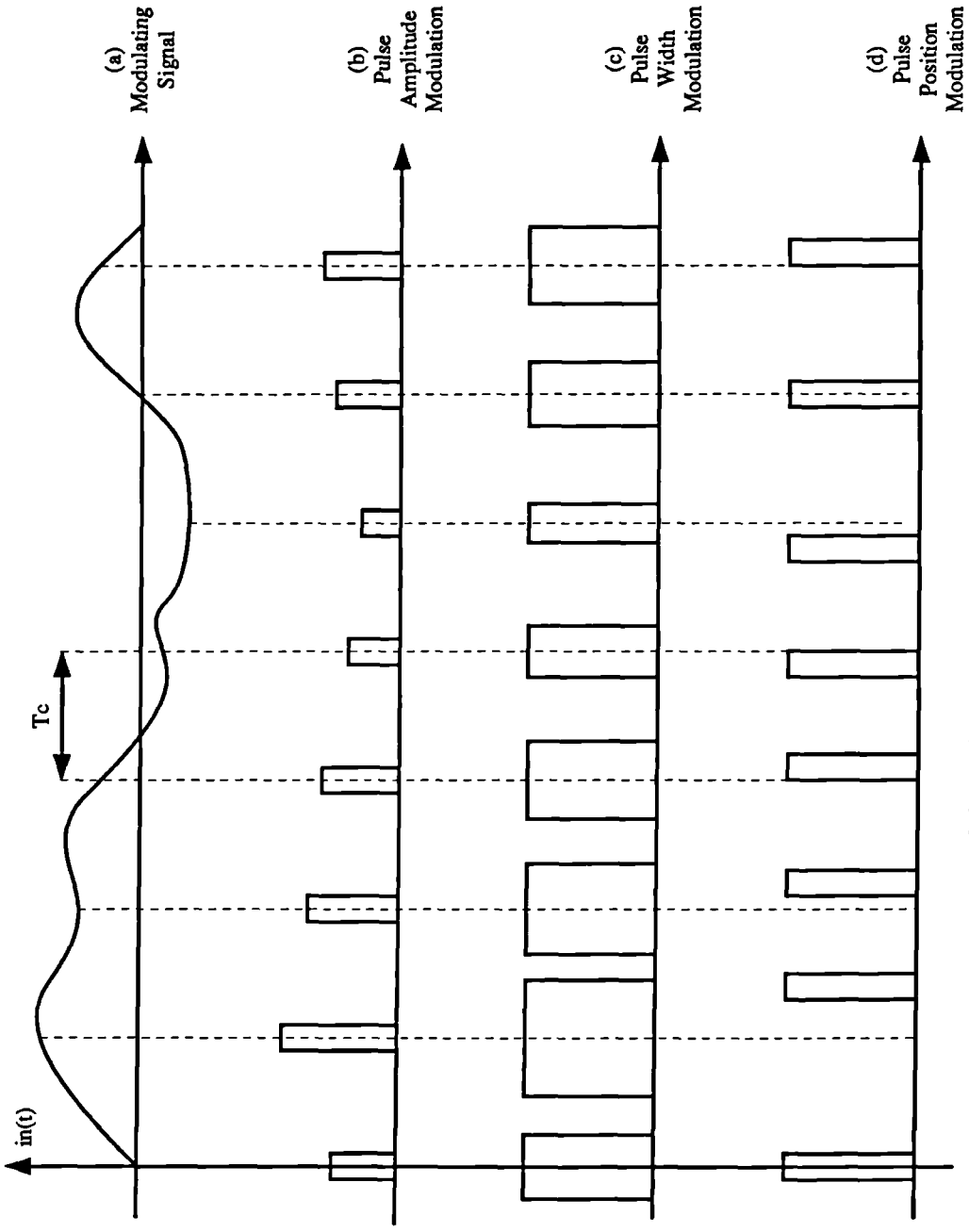


Fig. 2.1: Pulse Modulation Systems

of discrete output amplitude levels (as would be required when using PAM) [Hi92b].

Of the two PTM types, we have chosen PWM over PPM as the preferred modulation type. This is because the wider pulses associated with the former make it relatively easier to obtain adequately high levels of power amplification.

We begin this chapter with a description of the various classes of PWM. Then the tone spectra for several PWM modulation types are presented. These are discussed in detail. We also introduce approximations which make it easy to predict the relative size of the spectral components associated with each modulation type. Special issues connected with *digital* PWM are addressed, and guidelines for specifying system design parameters are given. We conclude with a discussion of the important nonidealities encountered in any practical implementation of PWM based DACs or PWM based digital power amplifiers.

2.2 Classification of PWM Modulation Types

As the name implies, PWM is a process whereby information-bearing signals are represented as variations in the width of high frequency pulses. As we shall see, the duration of each pulse is a function of the input signal amplitude at one or possibly two instants of time. A useful circuit analogy for ordinary, analogue PWM (presented in [Sa83, Sa86]) is shown in Fig. 2.2. Here the actual input signal, $in(t)$, or a zero order sample-and-held version, $in(nT_s)$, $n \in Z$ (Z denotes the set of all integers), is applied to the noninverting input of a comparator. (When the sample-and-hold is used it operates at a rate, $f_s \equiv 1/T_s$.) $cw(t)$, a high frequency comparison waveform with period T_c, is applied to the inverting input. The output of the comparator changes state such that a PWM waveform with "pulse repetition frequency," $f_c \equiv 1/T_c$, is produced.

Depending on whether the sample-and-hold device is used, different types of PWM can be generated. When it is bypassed "natural sampling PWM" (NPWM) results. If the sample-and hold is used, then $in(nT_s)$ is applied to the noninverting input of the comparator, and "uniform sampling PWM" (UPWM) is produced. Of course, the choice of comparison waveform also influences the type of PWM generated. If the comparison waveform is a simple sawtooth waveform then just one edge of each PWM pulse is modulated, resulting in "single sided PWM." This is shown in Fig. 2.3a-b where we consider the possibility of modulating the "trailing edge" of the pulse and that of modulating the "leading edge." When a triangle waveform is chosen both edges of the PWM pulse are modulated giving "double sided PWM." This is depicted in Fig. 2.4a. In general we note that

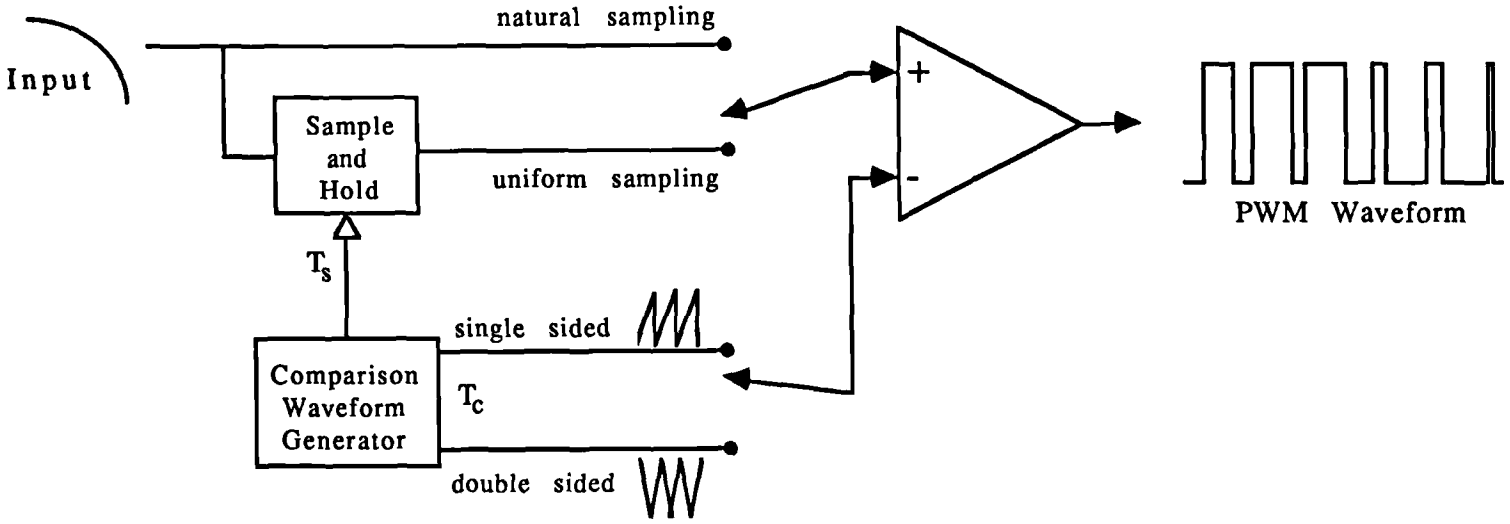


Fig. 2.2: Circuit Analogy for PWM

NPWM pulse widths are based on irregularly spaced samples with signal dependent sampling instants, while UPWM pulse widths are derived from regularly spaced samples of the input.

We see that for the two UPWM modulation types just described the pulse repetition frequency is equal to the sampling rate of the sample-and-hold device. However, this is not always the case. For double sided UPWM it is possible to modulate both edges of the PWM pulse with a single sample of the input signal (as shown in Fig. 2.4a) or to modulate the leading edge with one sample of the input and the trailing edge with the next sample. This gives rise to what is called "two sample consecutive" UPWM in [Sa83]. Note that *two* input samples are used in generating each pulse. Hence, two sample consecutive UPWM differs from those previously mentioned in that the pulse repetition frequency is half the sampling rate of the data. ($f_c=\frac{1}{2}f_s$). This is shown more clearly Fig. 2.4b.

While the input signals in the diagrams so far have been analogue signals, we are interested in applying amplitude quantized, digital signals to the modulator. This implies a type of *digital* PWM where the comparator of Fig. 2.2 becomes a digital comparator and the comparison waveform can be thought of as the output of a digital counter. For inputs quantized to b bits, there will be 2^b possible output pulse widths. In other words, the PWM pulse widths are quantized in time with the same resolution that the input to the modulator is quantized in amplitude. Hence, there is a minimum resolvable difference in pulse width corresponding to one least significant bit of the digital representation of the input signal. This presents us with the further option of generating symmetric or asymmetric pulses for the double sided, one sample per pulse UPWM waveform. In Fig. 2.5 we give a simple example where it is seen that the symmetric pulses require twice the time base resolution of the asymmetric waveforms. We also note from part b of the figure that the asymmetry only occurs for pulses of odd numerical length. In this case pulses can be generated such that the leading "half" of the pulse is always longer than the trailing "half," or vice versa, or we may choose to alternate in some manner which half is made longer.

Before concluding this section we note that all of the above single polarity (two level) modulation types are known collectively as "class AD" modulation types [Ma70]. There also exists a class of corresponding dual polarity (three level) "class BD" modulation types. The polarity of the class BD PWM pulse is determined by the polarity of the input signal at the sampling instant while its width is determined by the magnitude of the input signal at the same sampling instant. This is shown in Fig. 2.6 for both single sided and double sided UPWM. For digital power amplification, however, practical problems exist in the construction of high quality class BD power switching stages. (For more information on BD see [Bo75,Go90a,Go90b].) Specifically, aside from increasing circuit complexity,

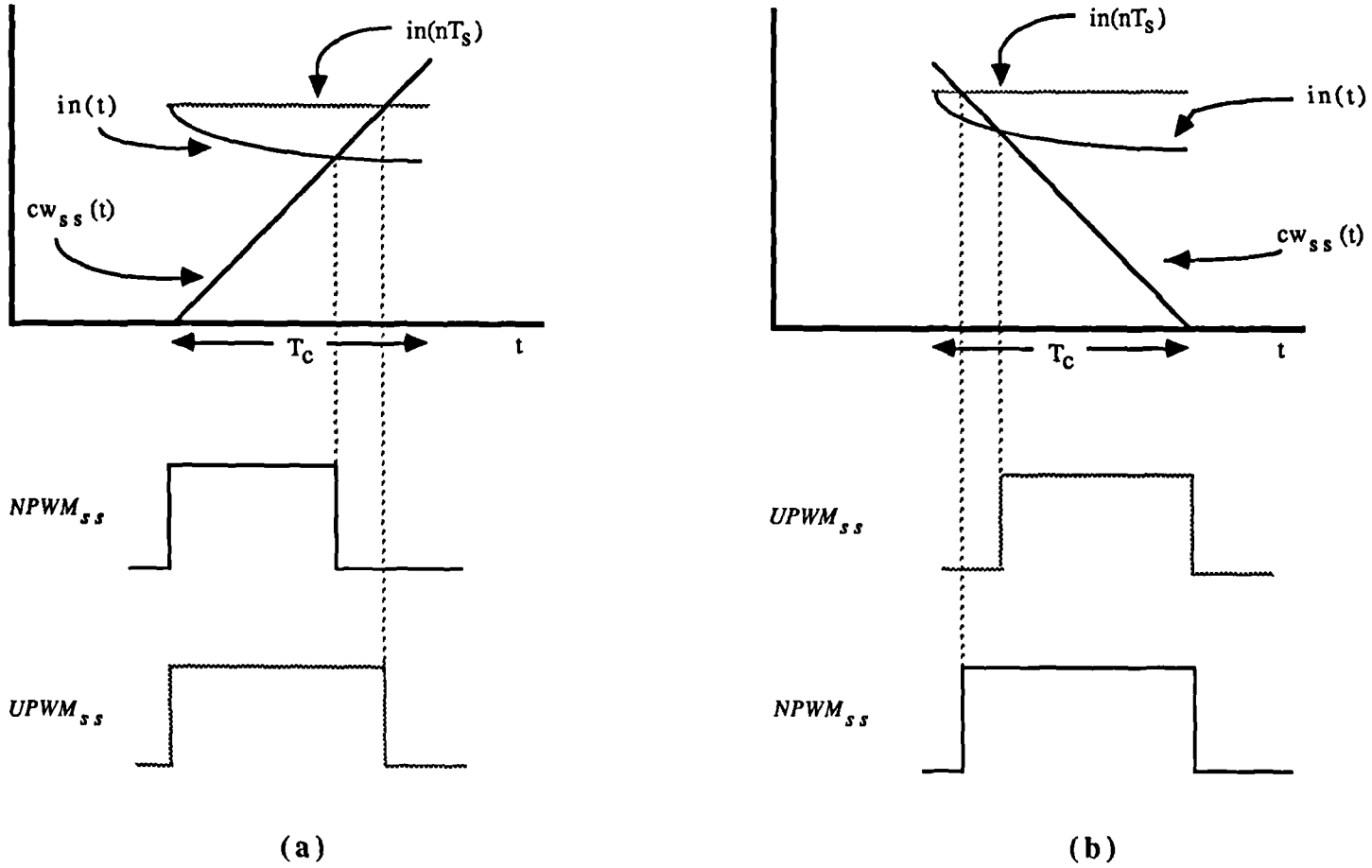


Fig: 2.3: Single Sided NPWM and UPWM
(a) trailing edge, (b) leading edge

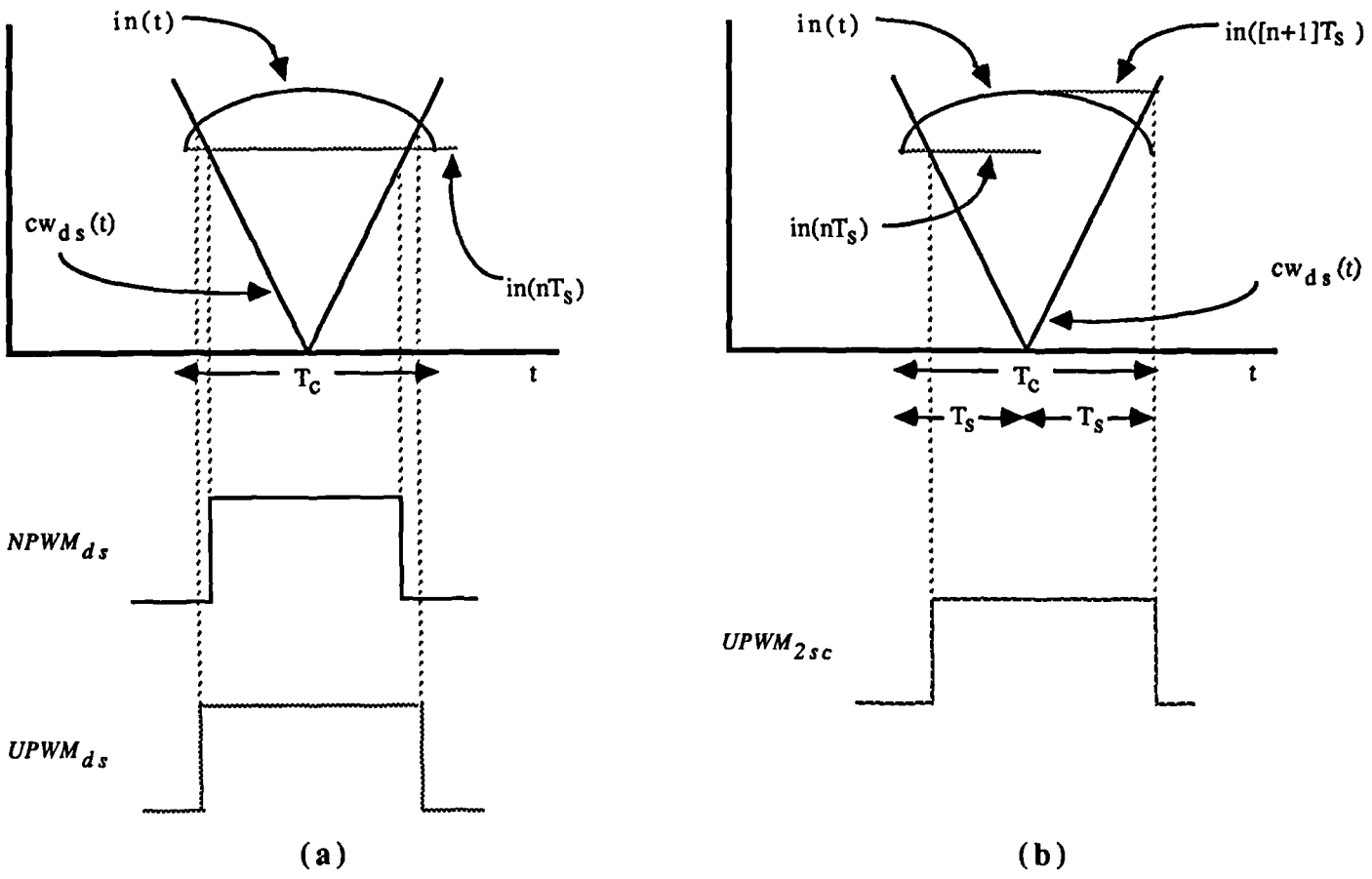
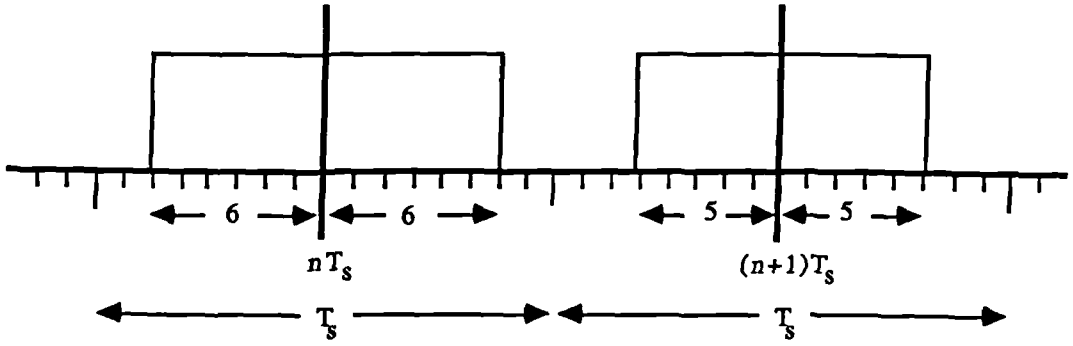
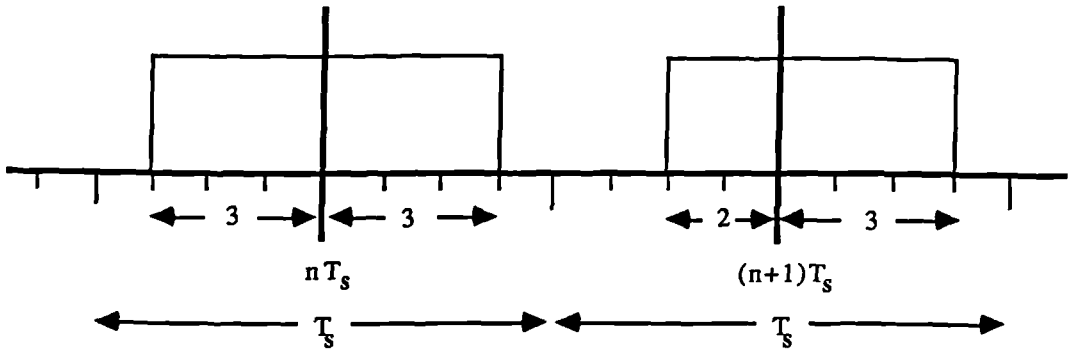


Fig 2.4: Double Sided NPWM and UPWM
(a) UPWM based on one sample ($T_c = T_s$)
(b) UPWM based on two samples ($T_c = 2T_s$)



regular timing markers correspond to half an LSB

(a)



regular timing markers correspond to a full LSB

(b)

Fig. 2.5: Double Sided UPWM Schemes (simple 3 bit example) (a) symmetric (b) asymmetric

there are difficulties in properly regulating the power supplies in an output stage with three or more levels [Hi92c]. For this reason we will restrict our discussion to class AD modulation types.

2.3 Analyses of PWM Modulation Types

The behaviour of the above types of pulse width modulators has been well understood for some time. In the early 1930's Bennett developed a technique for the analysis of a particular class of nonlinear modulation problems based on a double Fourier series expansion [Be33]. The technique has been used to analyze the performance of the five AD modulation types depicted in Figs. 2.2 to 2.4 for single tone inputs. [Bl53, Me91]. As the derivations are quite involved we refer the interested reader to the above references and simply state the results of the analysis for a single tone input, $M\cos\omega_0 t$.

2.3.1 Single Sided Modulation

The tone spectrum for single sided NPWM is given by [Bl53, Me91]:

$$NPWM_{ss}(t) = \tfrac{1}{2}M\cos\omega_0 t + \sum_{m=1}^{m=\infty}\frac{\sin(m\omega_c t)}{m\pi}\left[1+(-1)^{m+1}J_0(m\pi M)\right] + (-1)^{m+1}\sum_{m=1}^{m=\infty}\sum_{n=\pm1}^{n=\pm\infty}\frac{J_n(m\pi M)}{m\pi}\sin\left[m\omega_c t + n\omega_0 t - \tfrac{1}{2}n\pi\right] \tag{2.1}$$

where we have assumed that the ratio, for zero input, of output pulse width to the width of the interval between the centres of two consecutive pulses is one half. $M\in[0,1]$ is the modulation depth (i.e., the ratio of maximum pulse width to T_c), ω_0 is the angular frequency of the input tone, ω_c is the angular frequency of the carrier (i.e., the pulse repetition frequency) and $J_n(\cdot)$ is the nth order Bessel function of the first kind. We see that there are three groups of terms in the tone spectrum. There is an input tone term along with terms at the pulse repetition frequency and its harmonics in addition to sideband terms at multiples of the input tone frequency about the carrier frequency and its harmonics. This is shown in Fig. 2.7a.* The tone spectrum for single sided UPWM is [Me91]:

* The relative sizes of the spectral components are not to the correct scale.

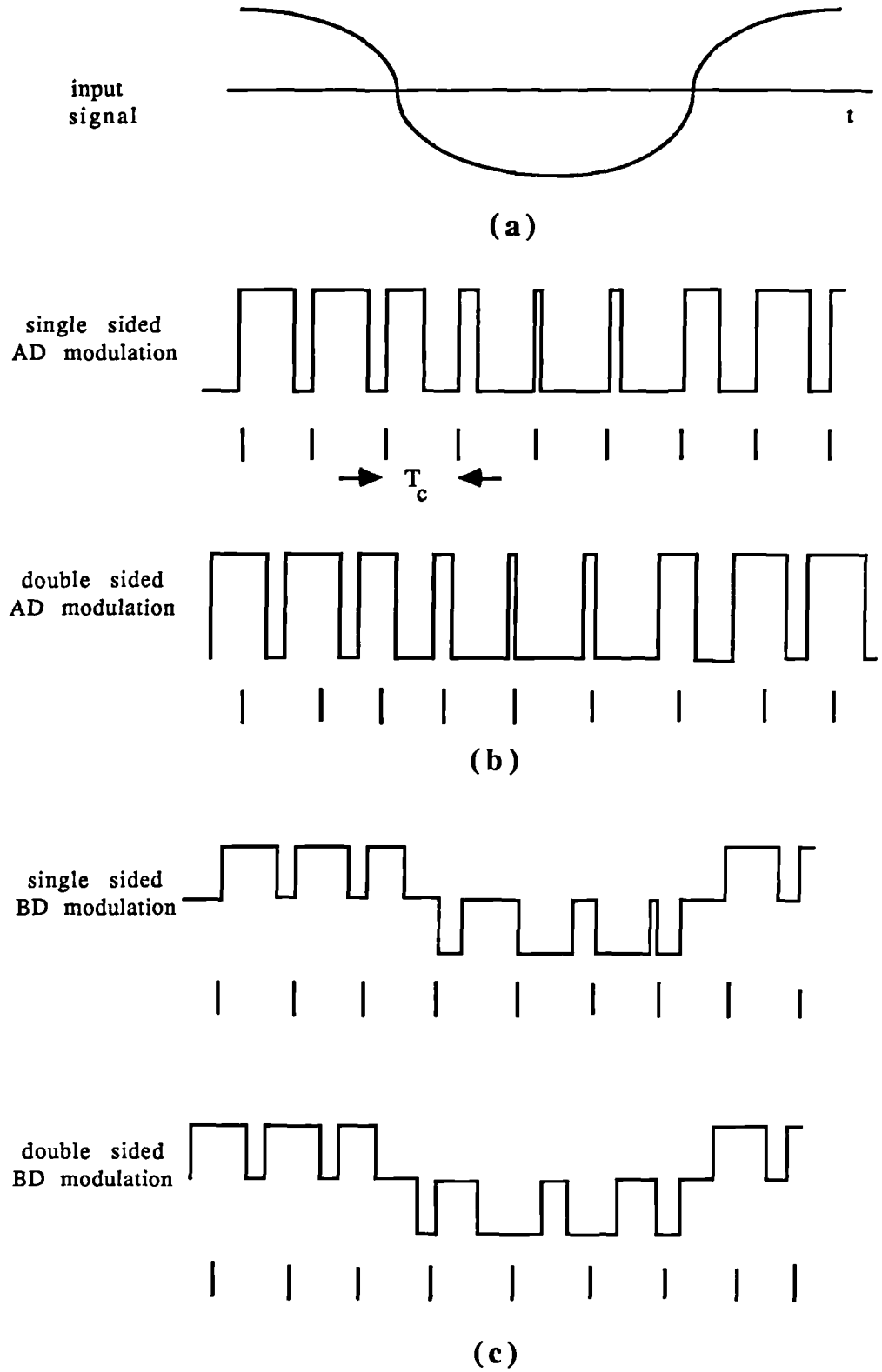


Fig. 2.6: PWM Schemes (Uniform)
(a) input; (b) class AD modulation;
(c) class BD modulation

$$UPWM_{ss}(t) = -\sum_{n=1}^{n=\infty} \frac{J_n(n\pi M\alpha)}{n\pi\alpha} \sin\left[n\omega_0 t - \tfrac{1}{2}\pi n(2\alpha+1)\right] \tag{2.2}$$

$$+ \sum_{m=1}^{m=\infty} \frac{\sin(m\omega_c t)}{m\pi}\left[1 + (-1)^{m+1} J_0(m\pi M)\right]$$

$$+ (-1)^{m+1} \sum_{m=1}^{m=\infty} \sum_{n=\pm 1}^{n=\pm\infty} \frac{J_n[\pi M(m+n\alpha)]}{\pi(m+n\alpha)} \sin\left[m\omega_c t + n\omega_0 t - \tfrac{1}{2}\pi n(1+2\alpha)\right]$$

where $\alpha = \omega_0/\omega_c$. Again there are three groups of terms with the components at the pulse repetition frequency and its harmonics identical to those of the NPWM spectrum in Eq. 2.1. However, an important difference is that there is an amplitude and phase distorted version of the input tone in addition to terms at the harmonics of the input. See Fig 2.7b.

2.3.2 Double Sided Modulation

We next present the double sided UPWM spectra. For the tone spectrum of double sided NPWM we have [Me91]:

$$NPWM_{ds}(t) = \tfrac{1}{2}M\cos\omega_0 t \tag{2.3}$$

$$+ 2\sum_{m=1}^{m=\infty} \frac{J_0(\tfrac{1}{2}m\pi M)}{m\pi} \sin(\tfrac{1}{2}m\pi)\cos(m\omega_c t)$$

$$+ 2\sum_{m=1}^{m=\infty} \sum_{n=\pm 1}^{n=\pm\infty} \frac{J_n(\tfrac{1}{2}m\pi M)}{m\pi} \sin\left[\tfrac{1}{2}\pi(m+n)\right]\cos\left[(m\omega_c + n\omega_0)t\right]$$

The tone spectrum for double sided UPWM is given by [Me91]:

$$UPWM_{ds}(t) = 2\sum_{n=1}^{n=\infty} \frac{J_n(\tfrac{1}{2}n\pi M\alpha)}{n\pi\alpha} \sin\left[\tfrac{1}{2}\pi(m+n[1-\alpha])\right]\cos(n\omega_0 t - n\alpha\pi) \tag{2.4}$$

$$+ 2\sum_{m=1}^{m=\infty} \frac{J_0(\tfrac{1}{2}m\pi M)}{m\pi} \sin(\tfrac{1}{2}m\pi)\cos(m\omega_c t)$$

$$+ 2\sum_{m=1}^{m=\infty} \sum_{n=\pm 1}^{n=\pm\infty} \frac{J_n[\tfrac{1}{2}\pi M(m+n\alpha)]}{\pi(m+n\alpha)} \sin\left[\tfrac{1}{2}\pi(m+n)(1-\alpha)\right]\cos\left[m\omega_c t + n\omega_0 t - n\alpha\pi\right]$$

Again the second group of terms are the same for uniform and natural sampling. It is also

apparent from the $\sin(\frac{1}{2}m\pi)$ factor in the second term of both expressions that all the even harmonics of the carrier are zero. Similarly, we notice in the third term of the naturally sampled case that the multiples of the sideband about the carrier and its harmonics are zero whenever $m+n$ is even. This is shown in Figs 2.7c-d.

The tone modulation spectrum for two sample consecutive UPWM is given by [Me91]:

$$UPWM_{2sc}(t) = 2\sum_{n=1}^{n=\infty} \frac{J_n(\frac{1}{2}n\pi M\alpha)}{n\pi\alpha} \sin(\frac{1}{2}\pi n)\cos(n\omega_0 t - \frac{1}{2}n\alpha\pi) \tag{2.5}$$

$$+ 2\sum_{m=1}^{m=\infty} \frac{J_0(\frac{1}{2}m\pi M)}{m\pi} \sin(m\pi\frac{1}{2})\cos(m\omega_c t)$$

$$+ 2\sum_{m=1}^{m=\infty}\sum_{n=\pm 1}^{n=\pm\infty} \frac{J_n[\frac{1}{2}M\pi(m+n\alpha)]}{\pi(m+\alpha n)} \sin\left[\frac{1}{2}\pi(m+n)\right]\cos\left[m\omega_c t + n\omega_0 t - \frac{1}{2}n\alpha\pi\right]$$

In contrast to both the previous UPWM modulation types only the odd harmonics of the input tone are present. We also see that, as in double sided NWPM, the multiples of the sideband about the carrier and its harmonics are zero whenever $m+n$ is even. See Fig. 2.7e.

2.4 Interpreting the PWM Tone Spectra

When considering PWM for use in a high quality DAC we must be concerned about the presence of any unwanted spectral components in the baseband, $\omega \in (-\omega_b, \omega_b)$ in addition to nonlinear magnitude and phase distortion of the input itself. From Eqs. 2.2, 2.4, and 2.5 it can be seen that UPWM results in distortion of the fundamental as well as baseband harmonic distortion whenever $\omega_0 < \frac{1}{2}\omega_b$ ($\omega_0 < \frac{1}{3}\omega_b$ for two sample consecutive UPWM). It is also apparent from Section 2.3 that baseband "foldback" distortion may arise from the sideband terms about the carrier and its harmonics when $|m\omega_c + n\omega_0| < \omega_b$ (expect when $m+n$ is even in double sided NPWM and in two sample consecutive UPWM). We begin by considering the distortion created by the three UPWM modulation types on the input tone itself.

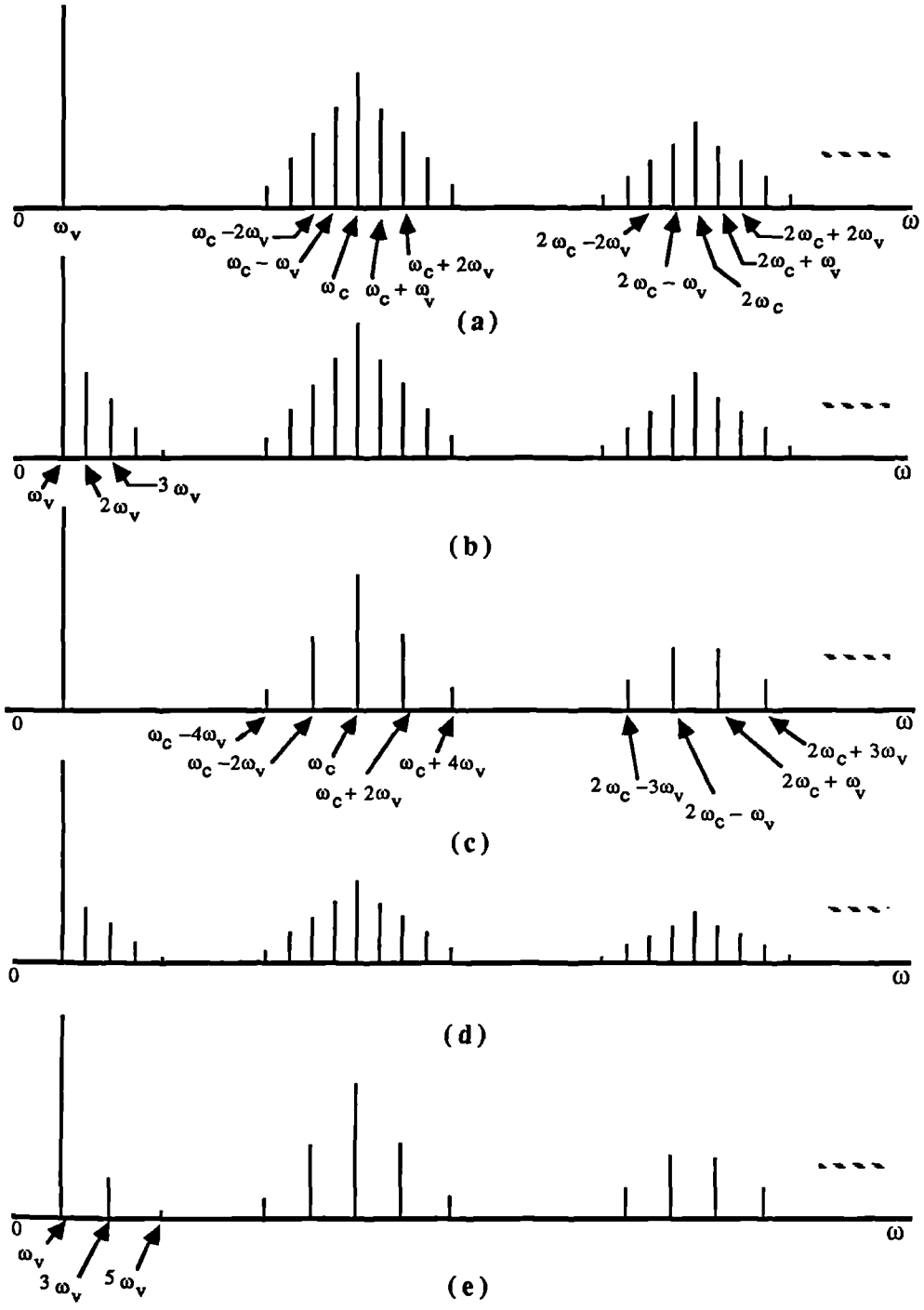


Fig. 2.7: Tone Spectra for Various PWM Types (not to scale)
(a) $NPWM_{ss}$ (b) $UPWM_{ss}$ (c) $NPWM_{ds}$ (d) $UPWM_{ds}$ (e) $UPWM_{2sc}$

2.4.1 Distortion of Fundamental (UPWM only)

Eqs. 2.2, 2.4, and 2.5 indicate that for all three UPWM modulation types the PWM output signal contains a magnitude and phase distorted version of the input. The magnitude of the output is weighted by a Bessel function which itself is a function of the input signal magnitude and frequency. In general, the distortion is of the form: $k_1 J_1(k_2 M x)/x$ where $x \propto \alpha (=\omega_y/\omega_c)$ where k_1 and k_2 are constants. In the double sided case of Eq. 2.4, there is an additional trigonometric term: $\sin\left[\frac{1}{2}\pi(1-\alpha)\right]$. It is known that [Me91]:

$$\lim_{\alpha \to 0} \frac{k_1 J_1(k_2 M \alpha)}{\alpha} = k_1 k_2 \frac{M}{2} \tag{2.6}$$

It is also easy to see that the trigonometric term approaches unity as α tends to zero. Recalling that the magnitude of the original input term is M, these facts indicate that for high pulse repetition frequencies (relative to signal frequency), nonlinear distortion of the magnitude may be quite small.

This is seen to be the case in Figs. 2.8a-b. Let F_1 denote the magnitude of the fundamental in the output of a UPWM waveform. Fig. 2.8a shows how this quantity (normalized to $\frac{1}{2}M$) changes with α, the ratio of signal frequency to pulse repetition frequency. As expected it indicates that the distortion is negligibly small for α small. Fig. 2.8b shows how $2F_1/M$ changes with M for α fixed at 0.5. Here we see that the distortion decreases as M becomes small. The double sided curve however tends to -3dB rather than 0dB. This is due to our choice of relatively large α (no oversampling) and its effect on $\sin[\frac{1}{2}\pi(1-\alpha)]$, which weights the double sided magnitude. (See Eq. 2.4.) The issue of magnitude distortion is considered again in Chapter Seven.

In all cases the phase of the output possesses a term directly proportional to the input frequency (i.e., a linear phase offset is introduced). Hence, phase distortion of the fundamental manifests itself benignly as a constant delay.

2.4.2 Harmonic Distortion (UPWM only)

General expressions for the levels of UPWM harmonic distortion relative to the input tone follow directly from Eqs. 2.2, 2.4, and 2.5 as ratios of Bessel functions and other trigonometric terms. They are shown in Table 2.1 where F_n denotes the magnitude of the nth harmonic of the input and

$$r \equiv M\alpha \quad \text{and} \quad \theta \equiv \tfrac{1}{2}\pi(1-\alpha) \tag{2.7,2.8}$$

Fig. 2.8a: UPWM Fundamental Distortion Levels vs alpha (M=1.0)

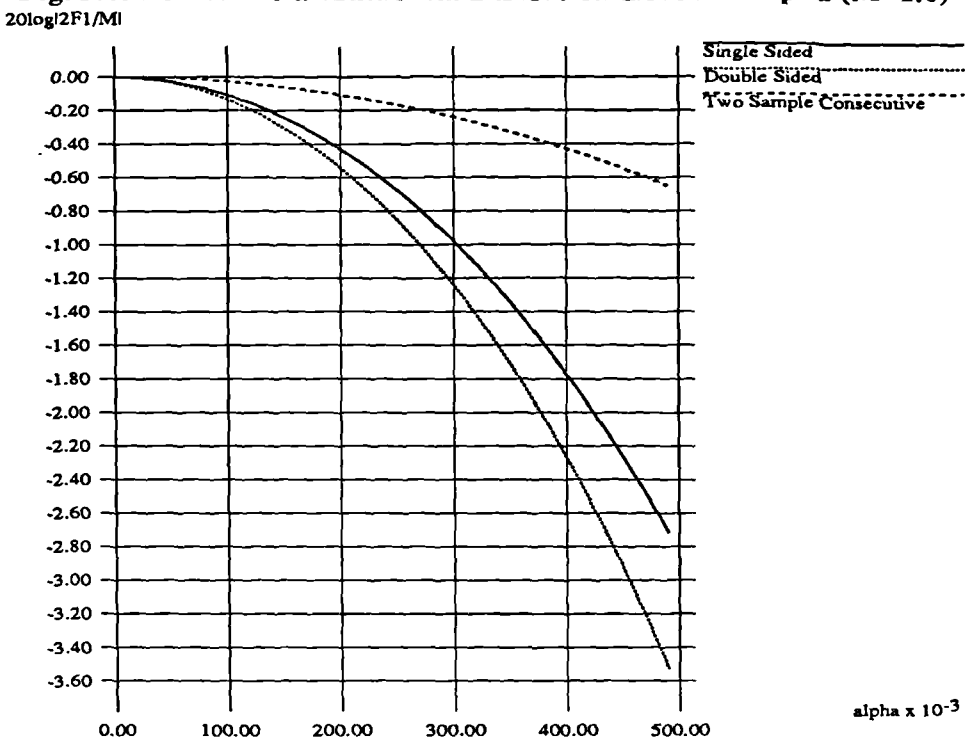


Fig. 2.8b: UPWM Fundamental Distortion Levels vs M (alpha=0.5)

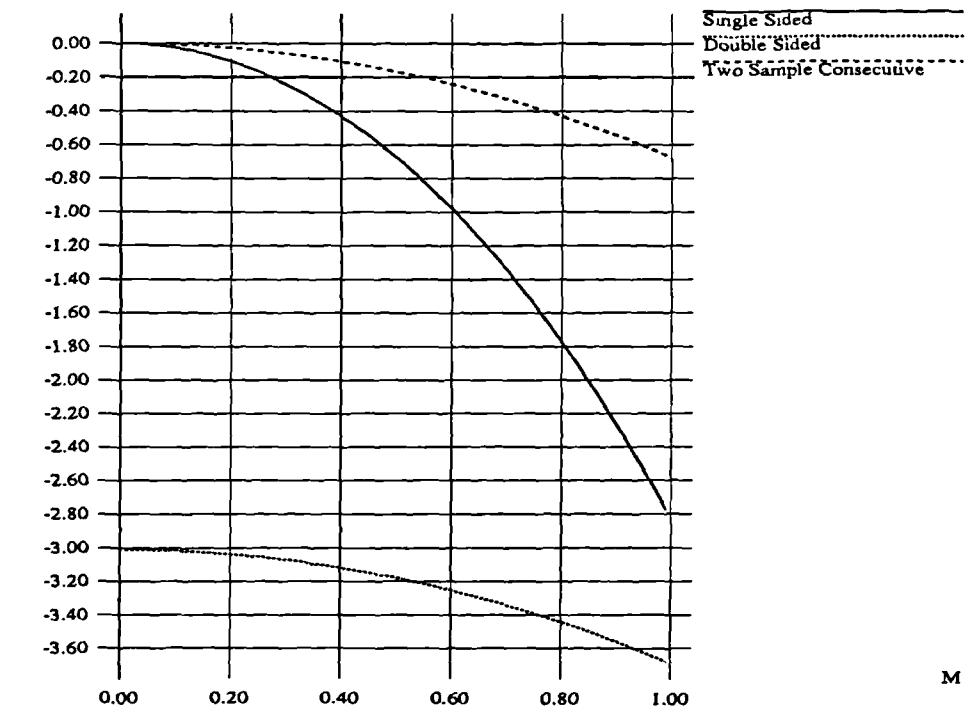


Fig. 2.9a: UPWM Harmonic Distortion Levels (2nd Harmonic) (M=1)

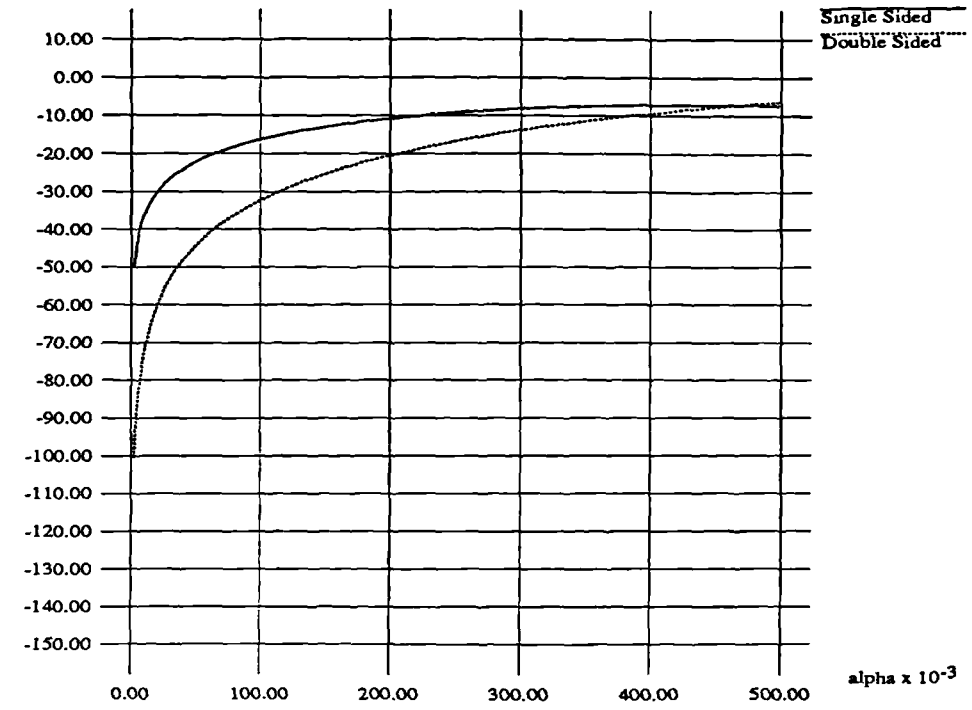


Fig. 2.9b: UPWM Harmonic Distortion Levels (3rd Harmonic) (M=1)

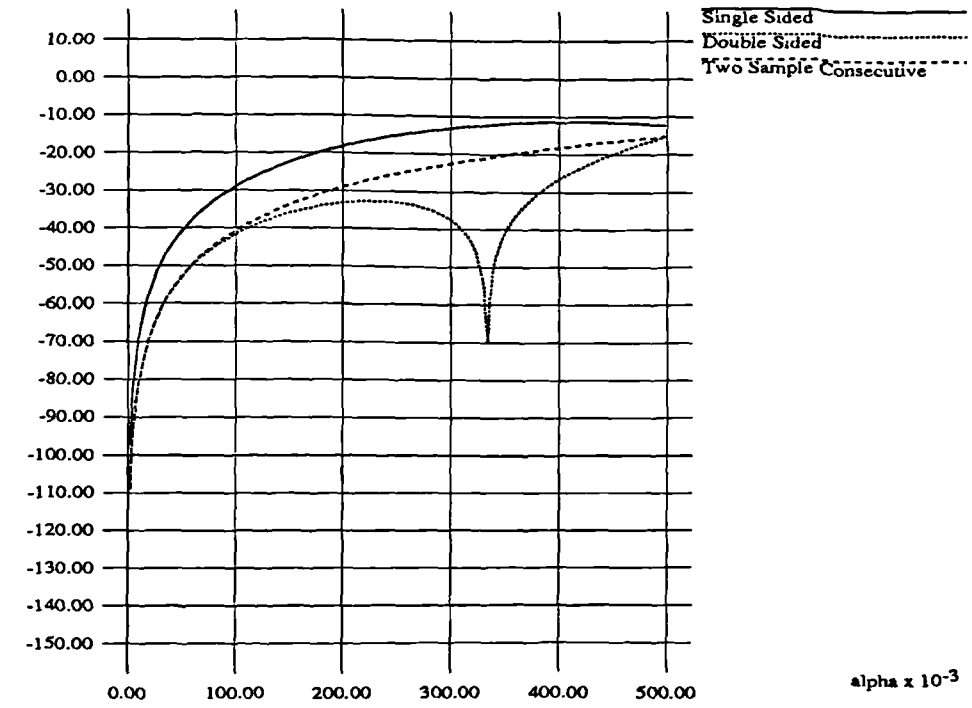


Table 2.1: UPWM Harmonic Distortion Levels (Relative to Fundamental)			
Distortion Level	Single Sided	Double Sided	Two Sample Consecutive
F_n/F_1	$\left\lvert \frac{J_n(\pi n r)}{n J_1(\pi r)} \right\rvert$	$\left\lvert \frac{J_n(\frac{1}{2}\pi n r)}{n J_1(\frac{1}{2}\pi r)} \sin(n\theta)/\sin(\theta) \right\rvert$	$\left\lvert \frac{J_n(\frac{1}{2}\pi n r)}{n J_1(\frac{1}{2}\pi r)} \sin(n\frac{1}{2}\pi) \right\rvert$

Graphs of the log magnitude levels of distortion as a function of ω_v/ω_c for the second to fifth harmonics are shown in Figs. 2.9a-d. In all four graphs the modulation depth, M, is set to unity, resulting in $r = \alpha = \omega_v/\omega_c$. Before examining the plots recall that we are particularly concerned with *baseband* distortion. This means that even for low pulse repetition frequencies, input tones over only a portion of the total input frequency range will result in harmonic distortion in the actual baseband, $\omega \in (-\omega_b, \omega_b)$. So for a given harmonic number n there is an upper bound on the ratio of input frequency to pulse repetition frequency of $\frac{\omega_b}{\omega_c n} < \frac{1}{2n}$ which will result in baseband nth order harmonic distortion. (This also places corresponding restrictions on the maximum value of r resulting in baseband harmonic distortion.) For instance, if ω_c=2.0, ω_b=0.5, and ω_v=0.2, only the second harmonic distortion term (at $2\omega_v$=0.4) would in fact be present in the baseband with the higher order distortion terms falling outside the baseband. In this example the maximum input frequency resulting in second order baseband harmonic distortion is $\frac{1}{2}\omega_b = 0.25$.

The plots indicate that for each modulation type nth order baseband harmonic distortion $\left[\text{i.e., } \alpha = \frac{\omega_v}{\omega_c} < \frac{1}{2n}\right]$ decreases as α is decreased. Thus *baseband* harmonic distortion is directly related to α. So for a given input frequency harmonic distortion can be reduced by increasing the pulse repetition frequency. Also, for the modulation type and α specified, we see that the level of harmonic distortion generally decreases as the harmonic number increases.* (Two sample consecutive UPWM is an exception due to the absence of even order harmonics). For each harmonic single sided UPWM tends to have higher levels of distortion than either the double sided or two sample consecutive modulation types. Also, the distortion associated with single sample double sided modulation is often less than or approximately equal to that of two sample consecutive modulation except for even harmonics where the latter produces no distortion. For small α the odd harmonic double sided and two sample consecutive modulation type curves approach one another and eventually become indistinguishable. In addition, we note that the presence of the unusual looking

* Experience has shown that for the 16 bit systems we consider in Chapter Seven baseband harmonics above n=4 are well below the quantization noise floor.

Fig. 2.9c: UPWM Harmonic Distortion Levels (4th Harmonic) (M=1)

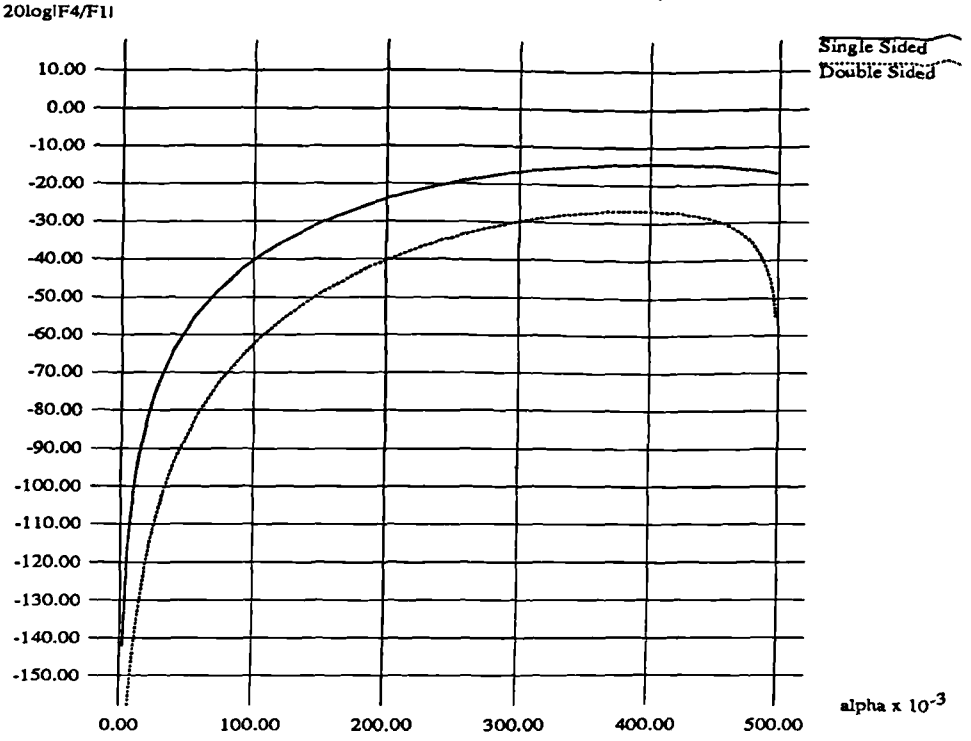
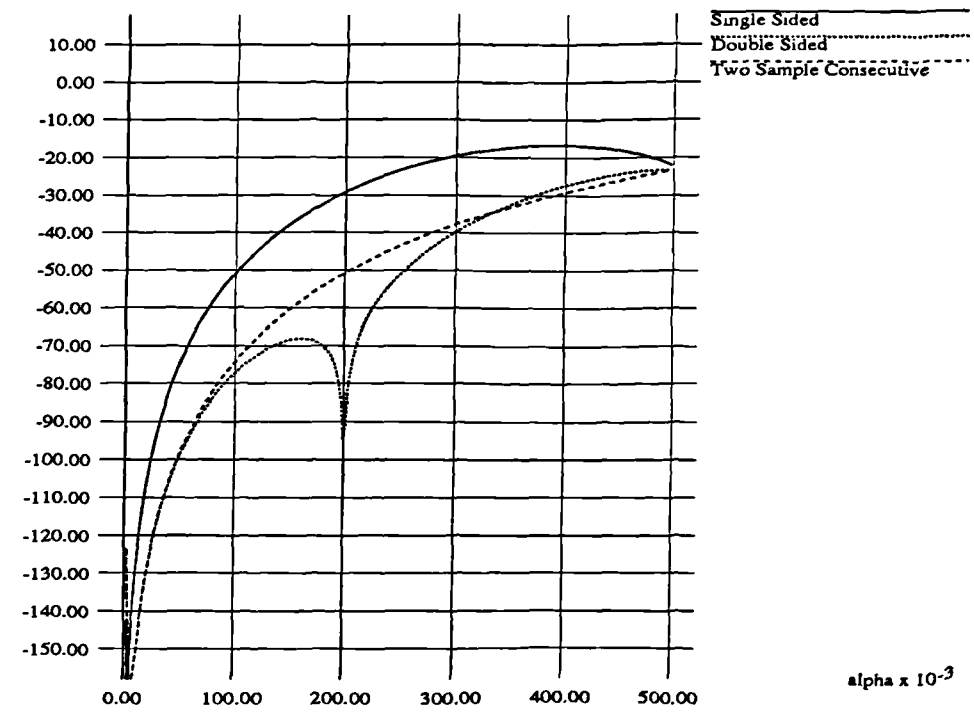


Fig. 2.9d: UPWM Harmonic Distortion Levels (5th Harmonic) (M=1)



dips in the plots for double sided modulation are due to the trigonometric term in the expression for the distortion level passing through zero. In all such cases the frequency of the harmonics at these dips numerically correspond to the pulse repetition frequency, ω_c, which is, of course, well above the baseband. More generally, we also see that while the largest harmonic distortion occurs for high frequency inputs, such distortion is often at frequencies above the baseband.

In Figs. 2.10a-d we also show the effect on the levels of harmonic distortion produced by varying the modulation depth, M, with $\alpha = \omega_v/\omega_c = f_v/f_c$ fixed at 0.25. At this α we see that for each modulation type harmonic distortion decreases as M is decreased. We again note that the distortion for a given modulation type decreases as the harmonic number increases.

The presence of the Bessel functions in the PWM spectra make it difficult to obtain accurate numerical values for the levels of the spectral components (such as harmonic distortion) without the use of tables and/or some computer algorithm for computing the Bessel functions. While the graphs in Figs. 2.9 and 2.10 are useful for gauging the levels of harmonic distortion, there is a more convenient approach. Since the Bessel function, $J_n(\rho)$, may be approximated by [Ol62]:

$$J_n(\rho) \approx \frac{1}{\sqrt{2\pi n}}\left[\frac{e\rho}{2n}\right]^n \quad \rho<<n, \tag{2.9}$$

easy-to-compute approximations to the levels of distortion can be derived. In our application, $\rho \propto r = M\alpha$ so the approximations are particularly accurate for small M (i.e., small input signal amplitude) and/or small α (i.e., low input signal frequency relative to pulse repetition frequency). In practice, some form of oversampling (i.e., increasing the PWM pulse repetition frequency above the Nyquist minimum) is used to reduce distortion. This implies that α will be small and that our approximations will be accurate.

These distortion levels are, as before, functions of r and θ. Using the approximation in Eq. 2.9 for the Bessel functions we can obtain estimates of the relative levels of the harmonic distortion associated with each of the three UPWM modulation types [Go91a,Sa86]. These are shown in Table 2.2.

Fig. 2.10a: UPWM Harmonic Distortion (2nd Harmonic) (alpha = 0.25)

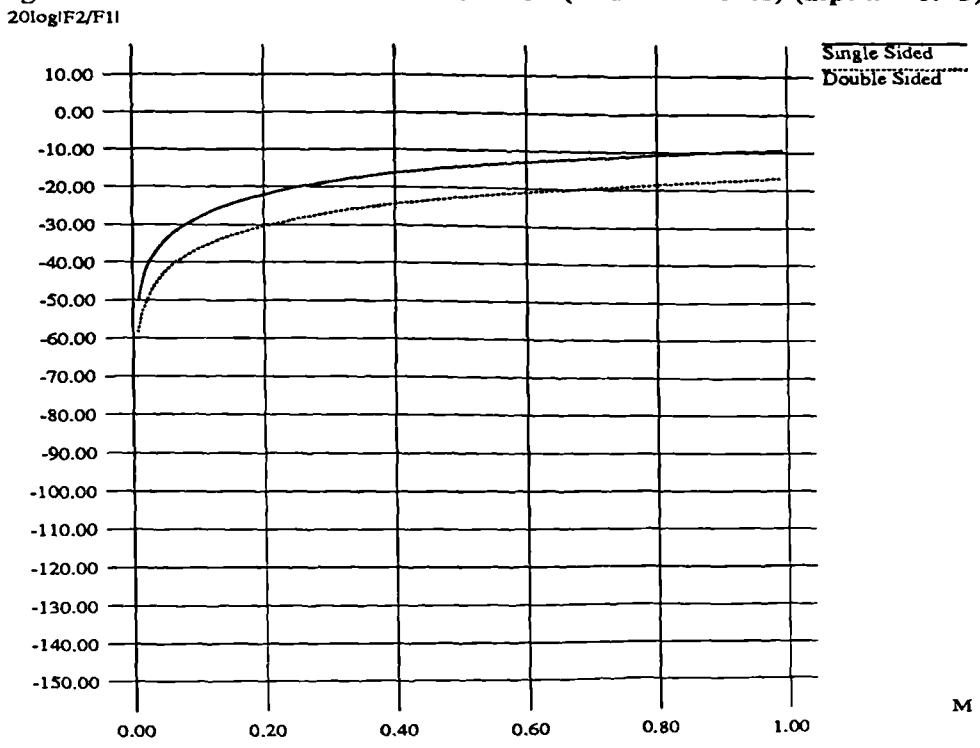


Fig. 2.10b: UPWM Harmonic Distortion (3rd Harmonic) (alpha = 0.25)

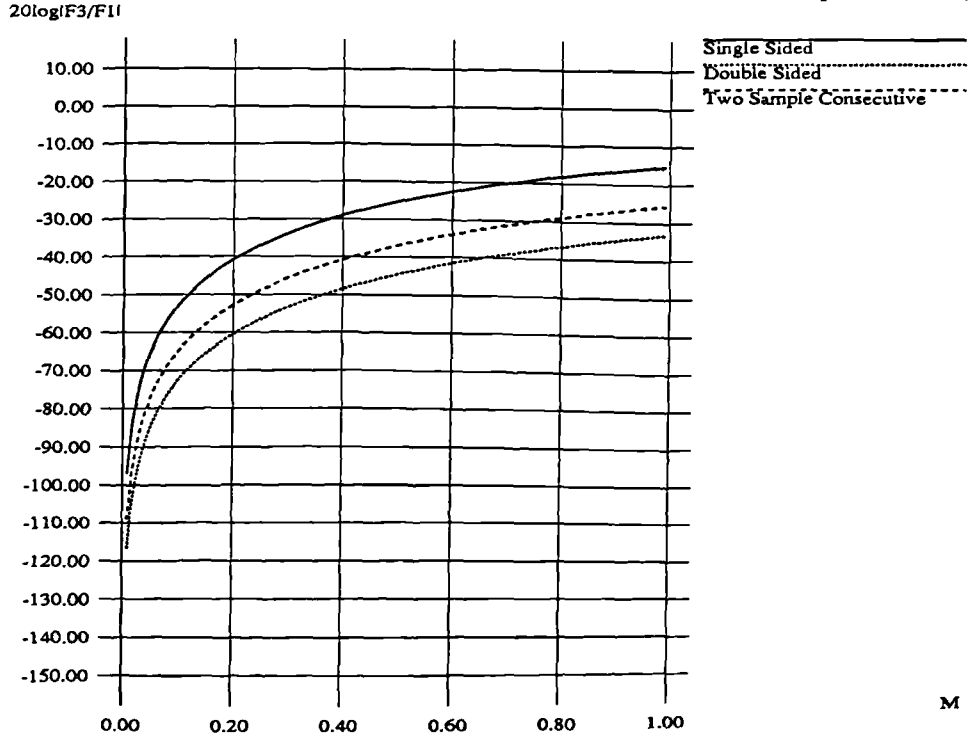


Fig. 2.10c: UPWM Harmonic Distortion (4th Harmonic) (alpha = 0.25)

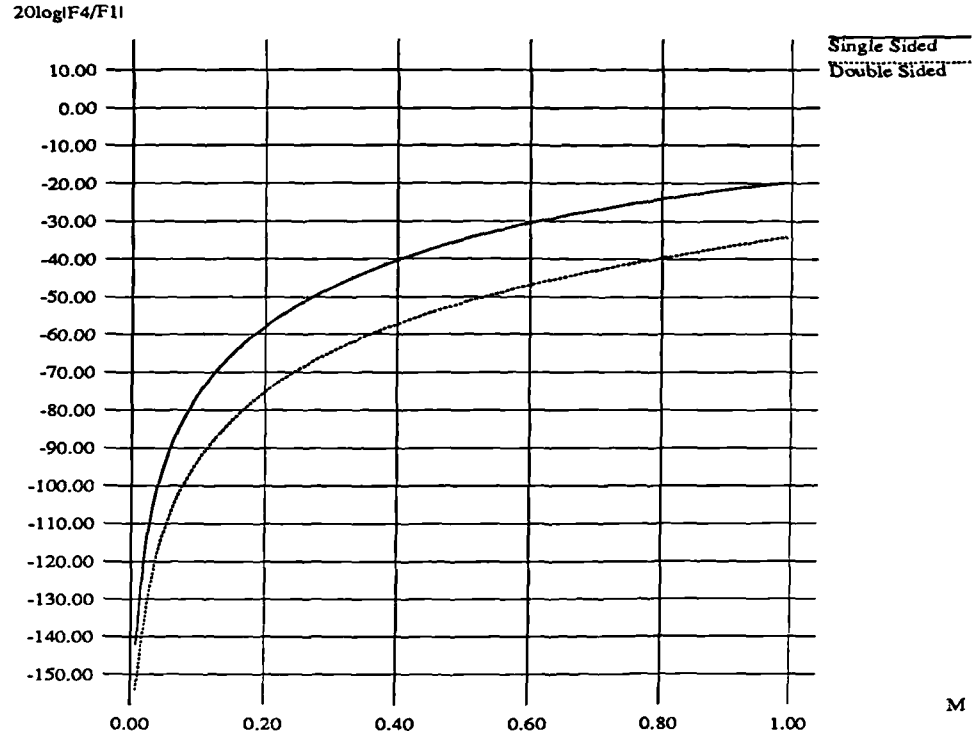


Fig. 2.10d: UPWM Harmonic Distortion (5th Harmonic) (alpha = 0.25)

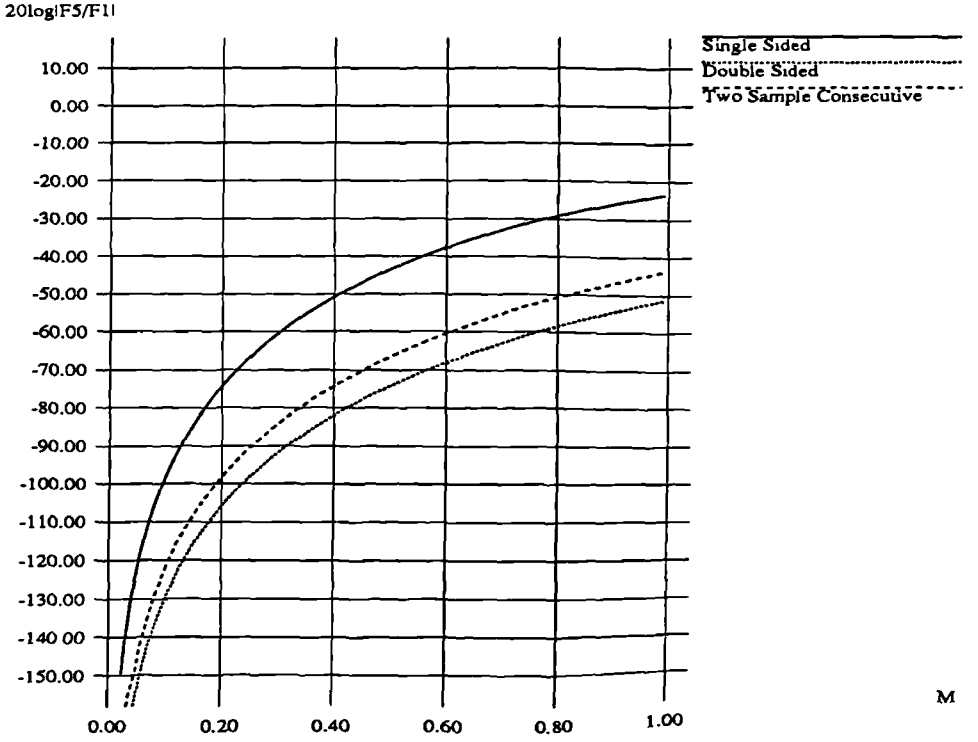


Table 2.2: UPWM Harmonic Distortion Level Approximations (Relative to Fundamental)				
UPWM modulation type	F_2/F_1	F_3/F_1	F_4/F_1	F_5/F_1
single sided	$1.51r$	$3.51r^2$	$9.73r^3$	$29.73r^4$
double sided	$1.51r\lvert\cos\theta\rvert$	$0.88r^2\left\lvert\frac{\sin 3\theta}{\sin\theta}\right\rvert$	$1.22r^3\left\lvert\frac{\sin 4\theta}{\sin\theta}\right\rvert$	$1.86r^4\left\lvert\frac{\sin 5\theta}{\sin\theta}\right\rvert$
two sample consecutive	-	$0.88r^2$	-	$1.86r^4$

The accuracy of these approximations improves when r is small and will (within a dB or so) predict the levels of harmonic distortion shown in Figs. 2.9 and 2.10 for $r<0.25$. It is interesting to look at the trigonometric terms in the double sided harmonic distortion expressions:

$$\frac{\sin(n\,\theta)}{\sin\theta} = \frac{\sin(n\,½\pi[1-\alpha])}{\sin(½\pi[1-\alpha])} \qquad n\in\{2,3,4,5\},\ \alpha\in[0,½) \tag{2.10}$$

With oversampling $\alpha\ll½$, and the denominator of the above expression tends to one while the numerator tends to zero for n even and ±1 for n odd. As could be expected, many of the claims made earlier in this section based on the harmonic distortion graphs can also be deduced from these approximations.

To sum up, the harmonic distortion associated with each modulation type can be decreased by either increasing ω_c or reducing M—both having the effect of decreasing r. We have also seen that the double sided and two sample consecutive modulation types tend to offer lower harmonic distortion levels than single sided UPWM. Due to the absence of even order harmonics, two sample consecutive tends to offer the lowest *total* harmonic distortion for the baseband restrictions on $\alpha=\omega_s/\omega_c$ given earlier in this section. We have seen that while the higher frequency tones can produce higher levels of harmonic distortion, this distortion often falls "harmlessly" outside the baseband. However, it should also be noted that for all three UPWM modulation types the input tone itself ($n = 1$) is modulated some amplitude distortion and that the severity of this effect increases with r.

2.4.3 Foldback Distortion (NPWM and UPWM)

As in the previous sub-section, general expressions for the "foldback" distortion arising from the sideband components about the carrier and its harmonics are readily apparent

from Eqs. 2.1-2.5. Inspection of these equations indicates that such distortion occurs in the UPWM *as well as* the NPWM modulation types and are again functions of Bessel functions and trigonometric terms. Expressions for the foldback distortion associated with each modulation type are shown in Table 2.3.

Table 2.3: PWM Foldback Distortion Levels (Relative to Fundamental)

Modulation Type		F_{m+n}/F_1
NPWM	single sided	$\left\lvert 2\frac{J_n(m\pi M)}{Mm\pi}\right\rvert$
	double sided	$\left\lvert 4\frac{J_n(\frac{1}{2}m\pi M)}{Mm\pi}\sin[\frac{1}{2}\pi(m+n)]\right\rvert$
UPWM	single sided	$\left\lvert \frac{J_n[\pi M(m+n\alpha)]}{J_1(\pi r)}\ \frac{\alpha}{m+n\alpha}\right\rvert$
	double sided	$\left\lvert \frac{J_n[\frac{1}{2}\pi M(m+n\alpha)]}{J_1(\frac{1}{2}\pi r)}\ \frac{\alpha}{m+n\alpha}\ \frac{\sin(\frac{1}{2}\pi m+n\theta)}{\sin(\theta)}\right\rvert$
	two sample consecutive	$\left\lvert \frac{J_n[\frac{1}{2}\pi M(m+n\alpha)]}{J_1(\frac{1}{2}\pi r)}\ \frac{\alpha}{m+n\alpha}\sin[\frac{1}{2}\pi(m+n)]\right\rvert$

Compared to the harmonic distortion expressions we see that these expressions are complicated by the presence of an additional parameter, m, the harmonic index for the pulse repetition frequency. Recall that we are particularly concerned with baseband foldback distortion which may occur when $|m\omega_c+n\omega_0| < \omega_b$ ($\omega_0 \le \omega_b < \frac{1}{2}\omega_c$, $m \in Z | m \ge 1$, $n \in Z$). This means that we can have several foldback distortion terms in the baseband arising from sideband components and their multiples "folding back" from the carrier (m=1), the second harmonic of the carrier (m=2), etc. This also implies that the values of n giving rise to foldback distortion in the baseband are very much a function of other parameters such as m, ω_c, ω_0, and ω_b. Hence the situation is different from that of harmonic distortion where experience (see Chapter Seven) has shown we could just present plots for the first few harmonics ($n \in \{2,3,4,5\}$) and be confident that we have satisfactorily characterized the baseband harmonic distortion. Therefore, in the case of foldback distortion, it would be misleading to present a specific set of plots and attempt to generalize from them. Moreover, in some cases the complicated interrelationship that arises between the overall level of foldback distortion and the parameters m, n, ω_0, ω_b, ω_c, and M makes it difficult to draw the type of detailed conclusions made earlier for harmonic distortion. For these

reasons it is felt that foldback distortion for the five modulation types is best understood on a case by case basis.

However, with the aid of approximations to the foldback distortion based on Eq. 2.9, we can say something. As before these approximations are more accurate when the arguments to the Bessel functions in Table 2.3 are small compared to its order, n. This is normally the case for baseband components (i.e., we can expect $|n|$ to be relatively large—even with modest oversampling). The approximations are shown in Table 2.4:

Table 2.4: PWM Foldback Distortion Level Approximations (Relative to Fundamental)

Modulation Type		F_{m+n}/F_1
NPWM	single sided	$\frac{1}{\sqrt{2\pi\lvert n\rvert}}\left[\frac{e}{\lvert n\rvert}\right]^{\lvert n\rvert}\left[\tfrac{1}{2}\pi M m\right]^{\lvert n\rvert-1}$
	double sided	$\frac{1}{\sqrt{2\pi\lvert n\rvert}}\left[\frac{e}{\lvert n\rvert}\right]^{\lvert n\rvert}\left[\tfrac{1}{4}\pi M m\right]^{\lvert n\rvert-1}\left\lvert\sin(\tfrac{1}{2}\pi[m+n])\right\rvert$
UPWM	single sided	$\frac{1}{\sqrt{\lvert n\rvert}}\left[\frac{1}{\lvert n\rvert}\right]^{n}(\tfrac{1}{2}e\,\pi M\,[m+n\,\alpha])^{\lvert n\rvert-1}$
	double sided	$\frac{1}{\sqrt{\lvert n\rvert}}\left[\frac{1}{\lvert n\rvert}\right]^{\lvert n\rvert}(\tfrac{1}{4}e\,\pi M\,[m+n\,\alpha])^{\lvert n\rvert-1}\left\lvert\frac{\sin(\tfrac{1}{2}\pi m+n\,\theta)}{\sin(\theta)}\right\rvert$
	two sample consecutive	$\frac{1}{\sqrt{\lvert n\rvert}}\left[\frac{1}{\lvert n\rvert}\right]^{\lvert n\rvert}(\tfrac{1}{4}e\,\pi M\,[m+n\,\alpha])^{\lvert n\rvert-1}\lvert\sin(\tfrac{1}{2}\pi[m+n])\rvert$

We begin with NPWM. For $|n|>1$ a direct relation between NPWM foldback distortion and M is readily seen. Also, it is apparent that double sided modulation tends to have lower foldback distortion than single sided modulation.

The situation for UPWM is more complicated. As with NPWM, there is a direct relation between UPWM foldback distortion and M. Also, the distortion for two sample consecutive modulation is less than that of single sided modulation. For double sided modulation first consider the trigonometric term:

$$\frac{\sin(\tfrac{1}{2}\pi m+n\,\theta)}{\sin(\theta)} = \frac{\sin(\tfrac{1}{2}\pi[m+n\,(1-\alpha)])}{\sin(\tfrac{1}{2}\pi[1-\alpha])} \tag{2.11}$$

For $\alpha\ll 1$ (which is often the case in *oversampled* DACs) the denominator of the above is near to (just slightly less than) unity while the numerator is either near to ± 1 for $m+n$ odd or near to zero for $m+n$ even. This means that in most cases the trigonometric term for

double sided UPWM foldback is less than or approximately equal to one. This further implies that the double sided foldback is less than or roughly equivalent to that of two sample consecutive UPWM foldback (except, of course, when $m+n$ is even forcing the latter to be identically zero). So, of the UPWM modulation types considered the double sided and two sample consecutive modulation types have lower foldback distortion than that of the single sided modulation type. Also for both single sided and double sided modulation, UPWM foldback distortion is often smaller than that of the corresponding NPWM modulation type (again with the $m+n$ even exception for double sided NPWM). To see this consider the term, $m+n\alpha$, which appears in the UPWM approximations. For baseband distortion:

$$|m+n\alpha| = \left| \frac{m\omega_c + n\omega_v}{\omega_c} \right| < \frac{\omega_b}{\omega_c} \tag{2.12}$$

With sufficient oversampling $m+n\alpha$ will be very small. It is the presence of these terms in the UPWM approximations which make UPWM foldback smaller than that of NPWM.

Lastly, as in the case of UPWM harmonic distortion, increasing ω_c is an effective means of reducing *baseband* foldback distortion for all five modulation types. This is seen to be the case by noting that for m and ω_b specified, an increase in ω_c often leads to an increase in the absolute value of n necessary to ensure that $|m\omega_c+n\omega_v|$ is indeed less than ω_b. In all five cases such an increase in $|n|$ usually leads to a large reduction in the level of baseband foldback distortion. (For similar reasons, in systems with ω_c significantly higher than the Nyquist minimum, the foldback distortion arising from multiples of the input tone about higher harmonics of the carrier ($m>1$) are usually negligibly small compared to the $m=1$ foldback.)

In summary, with both NPWM and UPWM, foldback distortion for double sided modulation is often less than that for single sided modulation. Also, for both single sided and double sided modulation, UPWM foldback is usually less than that of NPWM. Moreover, for all five modulation types, as with harmonic distortion, baseband foldback distortion can be reduced by increasing the pulse repetition frequency or reducing the modulation depth. (The former is often the preferred means of reducing distortion as large reductions in M can significantly reduce the output power of a digital amplifier.)

2.5 PWM Design Considerations

At this stage we ask how the above information can be used to design PWM systems to some quality specification. For example, given the input signal bandwidth, ω_b, what pulse repetition frequency, ω_c is required to ensure that the maximum baseband distortion component is less than some prescribed level? Consider first the harmonic distortion of the three UPWM modulation types. The worst case input (i.e., the tone input resulting in the highest baseband harmonic distortion) for single sided and double sided UPWM is a full scale tone at $\omega_0=\frac{1}{2}\omega_b$. ($\omega_0=\frac{1}{3}\omega_b$ for two sample consecutive UPWM.) Worst case scenarios for foldback distortion are often full scale tones at a frequency of ω_b folding back from the carrier (m=1)*. This is true because high frequency tones are folded back into the baseband with low $|n|$ which in general results in higher levels of distortion. With these worst case inputs, one can estimate from the approximations in Tables 2.3 and 2.4 the minimum pulse repetition frequency necessary to ensure that the distortion is less than a prescribed maximum level.

As an example, consider the design requirements for a system with a bandwidth of $f_b=\frac{\omega_b}{2\pi}=20kHz$ where no distortion component is to have power larger than -120dB with respect to the input signal. Starting with the two NPWM modulation types, use of the approximations in Table 2.4 yields conservative minimum pulse repetition frequencies of ~260kHz for single sided modulation and ~200kHz for double sided modulation. Next, consider the UPWM modulation types. Beginning by taking only foldback distortion into account, minimum pulse repetition frequencies of ~140kHz, ~120kHz, and ~120kHz are obtained for single sided, double sided, and two sample consecutive UPWM, respectively. When foldback distortion as well as harmonic distortion are taken into account we have ~15.1GHz, ~15.4MHz, and ~6.25MHz, respectively, all of which are much higher than the figures based only on foldback distortion. This is because baseband harmonic distortion levels decrease more slowly as a function of ω_c than baseband foldback distortion. Hardware limitations make it extremely difficult to realize digital amplifiers with pulse repetition frequencies as high as those quoted directly above for the UPWM modulation types. This implies that NPWM systems are particularly desirable in that high quality performance can be achieved at a reasonably low pulse repetition frequency. Other important

* The presence of the trigonometric terms in Table 2.4 can make determination of the worst case input tricky for the double sided and two sample consecutive modulation types. In these cases higher levels of distortion typically result when $|n|$ is relatively small, $|m+n\alpha|$ is relatively large, and $m+n$ is odd. In some instances these conditions are not met for an input signal of frequency $\omega_0=\omega_b$ but may be met by some other choice of ω_0.

limitations imposed by the hardware are discussed in the next section.

2.6 Non-idealities in PWM Circuits

A detailed investigation of the practical problems arising in hardware implementations of PWM based DACs and digital power amplifiers is now in progress [Hi92d]. In this section, we provide an overview of several of the ways in which the performance of an actual PWM circuit may deviate from the "idealized" versions assumed throughout this chapter (see[Hi91, Hi92e]).

First, power supply stability is an important issue. If the power supply is unstable and wobbles during a pulse, this will create distortion. Such instability can arise from noise or ripple from the mains power supply but can often be reduced by appropriate filtering or shielding.

Another problem is that of non-ideal pulse edges. Since all practical PWM circuits operate within some finite bandwidth and have nonzero rise and fall times, there will be some rounding of the ideal pulse edges shown in earlier in the chapter. If the rounding is the same for every pulse the overall effect can be modelled as a simple linear convolution of the ideal PWM waveform with some pulse edge shape function, $p(t)$. This has the effect of imposing a $P(\omega)$ function on the PWM spectrum where $P(\omega)$ is the Fourier Transform of $p(t)$. Usually $P(\omega)$ is a simple, gentle low pass type spectrum, and hence, as we are only concerned about the baseband, the effect is often not very severe. Nevertheless, in practice, guard band regions are used to ensure that there is adequate time for the switching transients to decay so as to avoid the PWM waveform turning on before it has finished turning off.

An additional issue is that of excessive modulator clock speed. The PWM circuits use a high frequency clock signal to time the pulse widths to the accuracy of the input signal. The clock speed required for a b bit system with pulse repetition frequency, $f_c=\omega_c/2\pi$, is:

$$f_{clk} = 2^b f_c \tag{2.13}$$

(This equation is for single sided systems in particular. For double sided symmetric and two sample consecutive UPWM the clock speed is an additional factor of two higher.) Here we see that clock speed increases exponentially as a function of wordlength. For a 16 bit signal with f_c=44.1kHz (i.e., standard digital audio specifications), f_{clk}=~2.89GHz. This figure will be even higher if guard bands and/or oversampling is used.

Another related problem is that of pulse edge time jitter due to noise in the clock circuits and variations in the characteristics of the logic circuits used in the modulator. This can be a serious problem since all the information about the modulating signal is carried in the time duration of the pulse. Fortunately though, for practical low power 16 bit quality applications pulse edge jitter has been shown to be low enough not to avoid serious problems [Hi91].

It is important to realize that for audio applications all of these problems are more severe when they are *signal dependent* (i.e., when the errors themselves are actually correlated to the width of the PWM pulse) as this is subjectively more disturbing.

2.7 Summary

In this chapter the basic PWM modulation types have been reviewed. Single sided and double sided NPWM and UPWM modulation types have been considered in detail. In particular, we have compared the harmonic distortion and the foldback distortion for tone inputs arising in each of these modulation types and have presented easy-to-compute approximations to the levels of distortion. We have seen how such distortion depends on input signal frequency, input signal bandwidth, pulse repetition frequency, and modulation depth. While most of the development was based on analogue PWM we have given some conceptual motivation for digital PWM as a digitized version of the original analogue circuit model. Lastly we discussed some of the nonidealities in PWM circuits which are likely to be encountered in any practical implementation of a PWM based DAC or power amplifier.

Chapter Three
Sample Rate Conversion

3.1 Introduction

As the name implies, "sample rate conversion" is the process by which the sampling rate of a digital signal is converted to some new rate. "Interpolation" and "decimation" are the names of the procedures used to increase and decrease, respectively, the sampling rate of a digital signal.

Digital sample rate conversion techniques are often employed in oversampling DACs and ADCs. Both interpolation and decimation are used in the simulation of the PWM based DACs considered in this thesis. In the previous chapter, we saw that increasing the pulse repetition frequency of a PWM waveform above the Nyquist minimum resulted in a reduction in baseband distortion. A digital PWM system with a high pulse repetition frequency requires a correspondingly high sampling frequency for the digital signal driving the modulator. Interpolation is used to increase the sampling rate of the Nyquist sampled input signal prior to modulation. Hence an interpolator would be part of any realistic hardware implementation of a PWM based DAC. (See Figs. 7.1, 7.9 and 7.29 of Chapter Seven.) On the other hand, a decimator is used in the simulation software to reduce the extremely high sampling rate of the sampled data approximation to the PWM waveform at the output of the software modulator. This is part of a narrow band spectral analysis procedure designed to accurately assess baseband performance of the DACs under simulation. As such, the decimator functions solely as an analysis tool and would not be part of the hardware implementation of a realistic PWM DAC.

This chapter summarizes the basics of digital interpolation and decimation techniques as used in the realization and computer simulation of PWM based DACs. We begin by showing interpolation and decimation as essentially complementary filtering operations. We then consider how these two procedures can be implemented more efficiently. Since the relevant theory is well established and, in our case, is applied in a standard manner, our

treatment of this subject will be brief. Readers interested in further details are referred to the comprehensive coverage offered in [Cr83].

3.2 The Basic Procedures

In this section we identify the basic interpolation and decimation procedures as dual digital low pass filtering operations.

3.2.1 Interpolation

Consider the task of raising the sampling rate of a discrete time signal, $x[n]$, by a factor, L, which is an integer greater than one. $x[n]$ may be thought of as arising from the sampling of a continuous time signal, $x_c(t)$, once every T seconds:

$$x[n] = x_c(nT) \tag{3.1}$$

The sampling rate associated with $x[n]$ is $F \equiv \frac{1}{T}$. As shown in Fig. 3.1, for the ideal case, the output of the interpolator is a signal, $y[m]$, which also corresponds to samples of $x_c(t)$ but with the smaller sampling interval, T':

$$y[m] = x_c(mT') \qquad T' = \frac{T}{L} \tag{3.2}$$

The sampling rate associated with $y[m]$ is

$$F' = \frac{1}{T'} = \frac{L}{T} = LF \tag{3.3}$$

Specifically how we get from $x[n]$ to $y[m]$ is shown in Fig. 3.2. $x[n]$ is first passed through a sample rate expander which interleaves $L-1$ zero valued samples between each sample in $x[n]$, obtaining a new signal $w[m]$:

$$w[m] = \begin{cases} x\left[\frac{m}{L}\right] & m = kL \quad k \in Z \\ 0 & m \neq kL \end{cases} \tag{3.4}$$

where Z is the set of all integers. This signal in turn is applied to a low pass filter to obtain the final output, $y[m]$.

It is useful to examine the procedure from the frequency domain. The magnitude of the Discrete Time Fourier Transforms of $x[n]$ and $y[m]$ are:

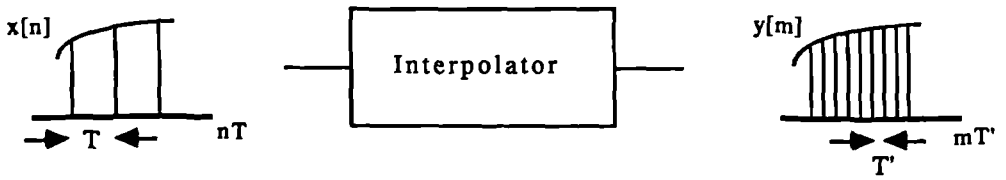


Fig. 3.1 Interpolation

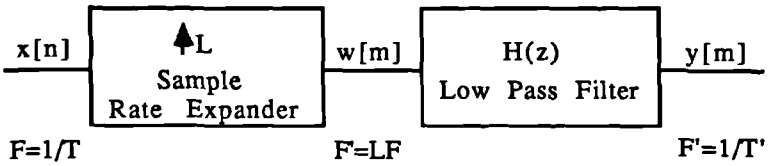


Fig. 3.2 Block Diagram for L Times Interpolation

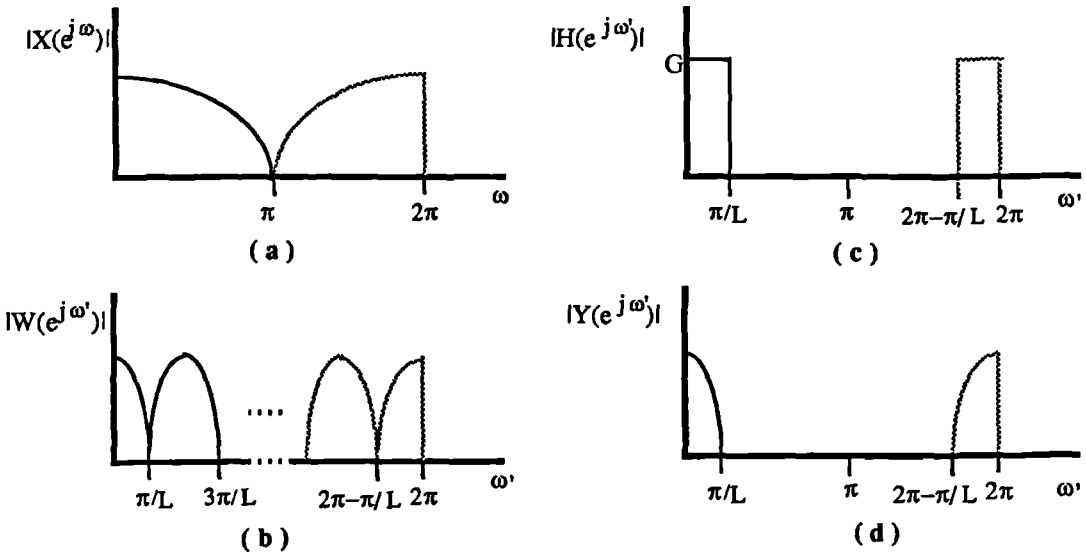


Fig. 3.3 Spectra for L Times Interpolation

$$\left|X(e^{j\omega})\right| \equiv \left|\sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}\right| \tag{3.5}$$

$$\left|Y(e^{j\omega'})\right| \equiv \left|\sum_{m=-\infty}^{\infty} y[m]e^{-j\omega' m}\right| \tag{3.6}$$

respectively, with normalized frequency variables,

$$\omega = 2\pi\frac{f}{F} \tag{3.7}$$

$$\omega' = 2\pi\frac{f}{F'} = 2\pi\frac{f}{LF} = \frac{\omega}{L} \tag{3.8}$$

The spectra are shown in Figs. 3.3a and 3.3d. It is can be shown that:

$$W(e^{j\omega'}) \equiv \sum_{m=-\infty}^{\infty} w[m]e^{-j\omega'} = X(e^{j\omega' L}) \tag{3.9}$$

which as seen in Fig. 3.3b implies that the sample rate expander while changing the sampling rate of the input signal leaves the shape of its spectrum unaltered. To obtain $y[m]$ from $w[m]$ it is necessary to remove the spectral images centred at $\omega' = \frac{2\pi}{L}, \frac{4\pi}{L}, \cdots, 2(L-1)\frac{\pi}{L}$. This is done by applying $w[m]$ to a low pass filter, $H(z)$, with ideal magnitude frequency response:

$$|H(e^{j\omega'})| = \begin{cases} G & \omega' \in \left[0, \frac{\pi}{L}\right] \\ 0 & \omega' \in \left[\frac{\pi}{L}, \pi\right] \end{cases} \tag{3.10}$$

as seen in Fig. 3.3c. Therefore, in the ideal case, the spectrum of the output can be written as:

$$|Y(e^{j\omega'})| = |H(e^{j\omega'})||X(e^{j\omega' L})| = \begin{cases} GX(e^{j\omega' L}) & \omega' \in \left[0, \frac{\pi}{L}\right] \\ 0 & \omega' \in \left[\frac{\pi}{L}, \pi\right] \end{cases} \tag{3.11}$$

It can be shown that to ensure correct scaling, G, the pass band gain of the filter, should be set to the interpolation factor, L.

For a realistic filter, there would be ripples in the pass band and finite attenuation of the spectral images in the stopband.

3.2.2 Decimation

Next, consider the task of lowering the sampling rate of a digital signal, $x[n] = x_c(nT)$ with sampling rate, $F=\frac{1}{T}$, by an integer factor, M, as shown in Fig. 3.4. As before, in the ideal case, the output of the decimator corresponds to a resampling of the original continuous time waveform at the new rate:

$$y[m] = x_c(mT') \tag{3.12}$$

where $T' = MT$. The new lower sampling rate is:

$$F' = \frac{1}{T'} = \frac{1}{MT} = \frac{F}{M} \tag{3.13}$$

The details are shown in Fig. 3.5. The input signal is first low pass filtered, creating an intermediate signal, $v[n]$, which is then applied to a sample rate compressor yielding the output, $y[m]$. The sample rate compressor simply passes every Mth sample of the output of the low pass filter:

$$y[m] = v[Mm] \qquad m \in Z \tag{3.14}$$

Again we view the procedure from the frequency domain. The spectra of the input and output signals are shown in Fig. 3.6a and d with normalized frequency variables, ω as in Eq. 3.7, and

$$\omega' = 2\pi\frac{f}{F'} = 2\pi\frac{f}{F}M = \omega M \tag{3.15}$$

Now, in general, the input to the decimator will possess baseband spectral content all the way up to half its sampling frequency, $\omega = \pi$ ($f = \frac{1}{2}F$). So, in order to prevent aliasing when the sampling rate of the signal is reduced, low pass filtering is required. The bandwidth of $y[m]$ will extend to $\omega' = \pi$ ($f = \frac{1}{2}F'$) or $\omega = \frac{\pi}{M}$ $\left[f = \frac{1}{2}\frac{F}{M}\right]$. As such, the ideal low pass filter will possess frequency response:

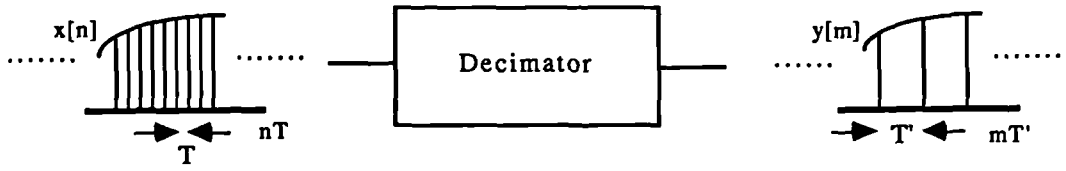


Fig. 3.4 Decimation

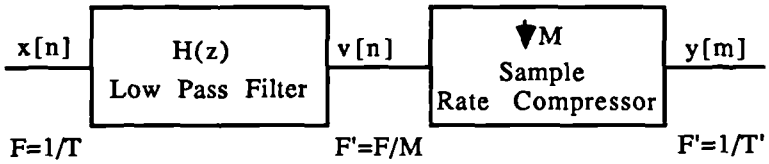


Fig. 3.5 Block Diagram for M Times Decimation

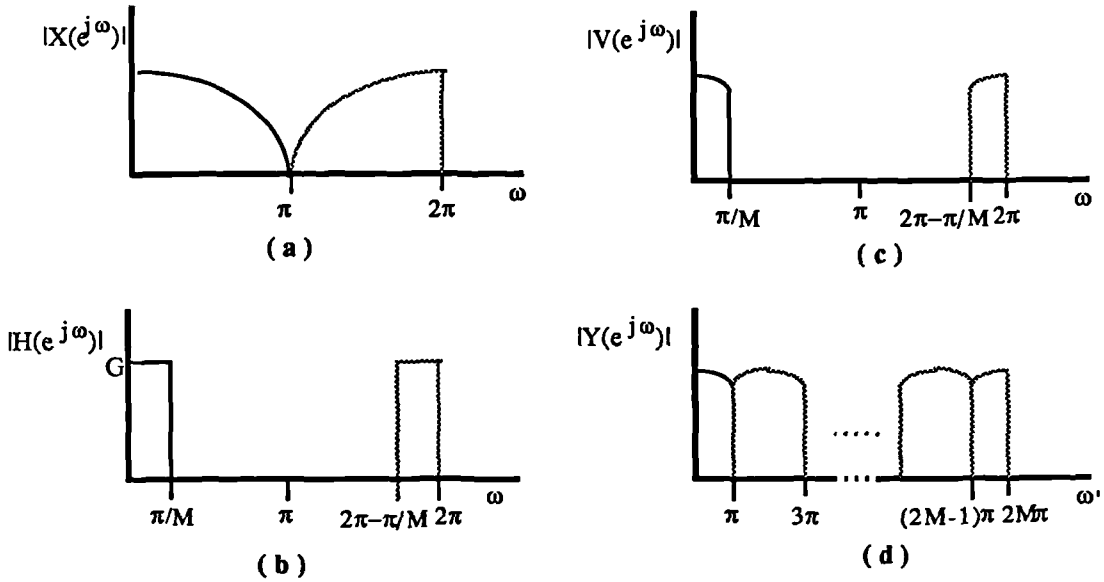


Fig. 3.6 Spectra for M Times Decimation

$$|H(e^{j\omega})| = \begin{cases} G & \omega \in \left[0, \frac{\pi}{M}\right] \\ 0 & \omega \in \left[\frac{\pi}{M}, \pi\right] \end{cases} \tag{3.16}$$

This is shown in Fig. 3.6b. Thus, ideally the spectrum of the output of the filter as shown in Fig. 3.6c is given as:

$$|V(e^{j\omega})| = |H(e^{j\omega})||X(e^{j\omega})| = \begin{cases} G\,|X(e^{j\omega})| & \omega \in \left[0, \frac{\pi}{M}\right] \\ 0 & \omega \in \left[\frac{\pi}{M}, \pi\right] \end{cases} \tag{3.17}$$

where in this case correct scaling can be shown to be obtained with G=1. It can also be shown that the spectrum of the output is related to that of the input by:

$$Y(e^{j\omega'}) = \frac{1}{M}\sum_{k=0}^{M-1} V(e^{j(\omega'-2\pi k)/M}) = \frac{1}{M}\sum_{k=0}^{M-1} H(e^{j(\omega'-2\pi k)/M}) X(e^{j(\omega'-2\pi k)/M}) \tag{3.18}$$

In the ideal case, where there will be no chance of aliasing, the above simply reduces to:

$$Y(e^{j\omega'}) = \frac{1}{M} X(e^{j\omega'/M}) \tag{3.19}$$

Again with realistic filters we should expect some pass band ripple as well as finite stop-band attenuation—the latter of which would give rise to a certain degree of aliasing.

3.3 Strategies for Improving Computational Efficiency

Now that we have established the basic ideas behind interpolation and decimation we survey some of the techniques used to decrease the computational complexity of these procedures. Of course, computational efficiency in the interpolator is important to help minimize the cost and complexity of the overall DAC. It is also very important for the decimator in the analysis software. In the simulation the signal at the output of the modulator is a sampled data approximation to the continuous PWM waveform. This approximation has a sampling rate several orders of magnitude higher than the Nyquist rate of the original input to the system. Decimation factors between $2^{11} = 2048$ and $2^{20} = 1048576$ are not uncommon. So to reduce the amount of time required to perform the analysis the software decimator should be implemented as efficiently as possible. In this section, we consider three ways in which the overall computational complexity of a sample rate

conversion system can be reduced. Specifically, we see how (i) the elimination of unnecessary operations in the basic procedure, (ii) the cascading of several stages of sample rate conversion, and (iii) the application of special classes of filters can all be used to reduce total levels of computational complexity.

3.3.1 Efficient Sample Rate Change Structures

We consider interpolation first. Recall that $w[m]$, the input to the low pass filter of Fig. 3.2, consists of $L-1$ zeros interleaved between the original input samples. Therefore, $L-1$ of out every L multiply/adds in the low pass filtering simply results in zero—making no contribution to the output of the interpolator. Hence, these $L-1$ out of L multiply/adds need not be performed. This reduces the computational complexity of the procedure by a factor of L and implies that in effect the computation can be performed at the low sampling rate. This can also be understood graphically from Fig. 3.7 for a transposed direct form FIR filter.* We see that the filter coefficients are commuted with the sample rate expander such that the multiplies take place at the low rate, F. In Fig. 3.7d the structure is further modified to exploit symmetry in the impulse response of the filter to obtain an additional factor of two reduction in the number of multiplies required.

Similar savings can be obtained in the decimator of Fig. 3.5 where $M-1$ out of every M output samples of the low pass filter are simply discarded by the sample rate compressor. A factor of M reduction in the computational complexity can be obtained by not performing the multiply/adds in the low pass filter for the samples which would be discarded anyway. This is shown in Fig. 3.8 for a (non-transposed) direct form FIR filter. The savings are the result of commuting the sample rate compressor with the filter coefficients, allowing the computation to take place at $F' = F/M$, the low output sampling rate. Again an additional factor of two reduction in the number of multiplies can be obtained by exploiting the symmetry of the impulse response of the filter.

* Throughout the remainder of this chapter we assume the use of FIR filters. This is because in our application we desire an overall linear phase (constant group delay) response. Also, the computational savings described in this section are most easily obtained with FIR filters.

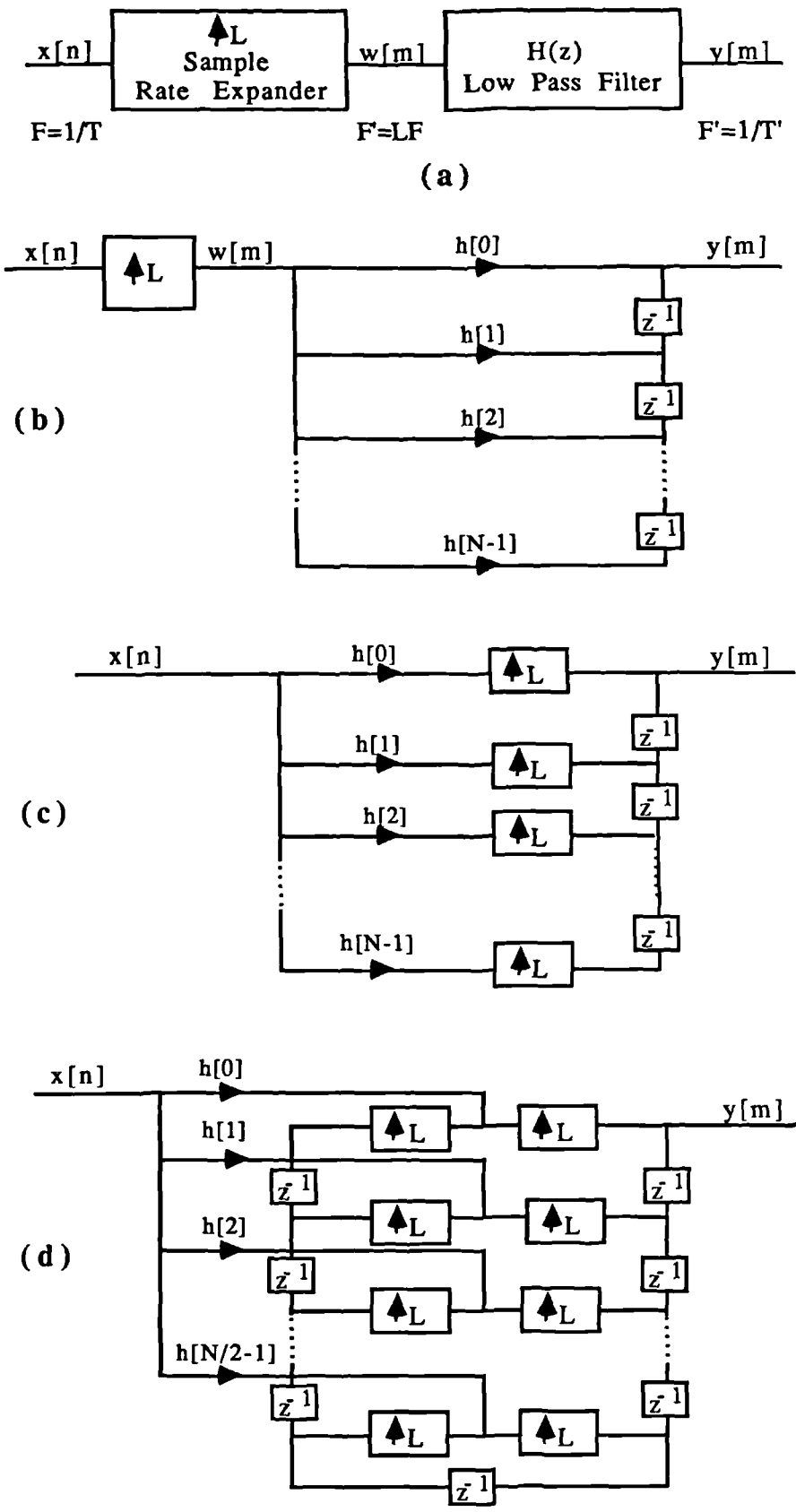


Fig. 3.7: Obtaining Efficient Interpolator Structures (assuming N even for (d))

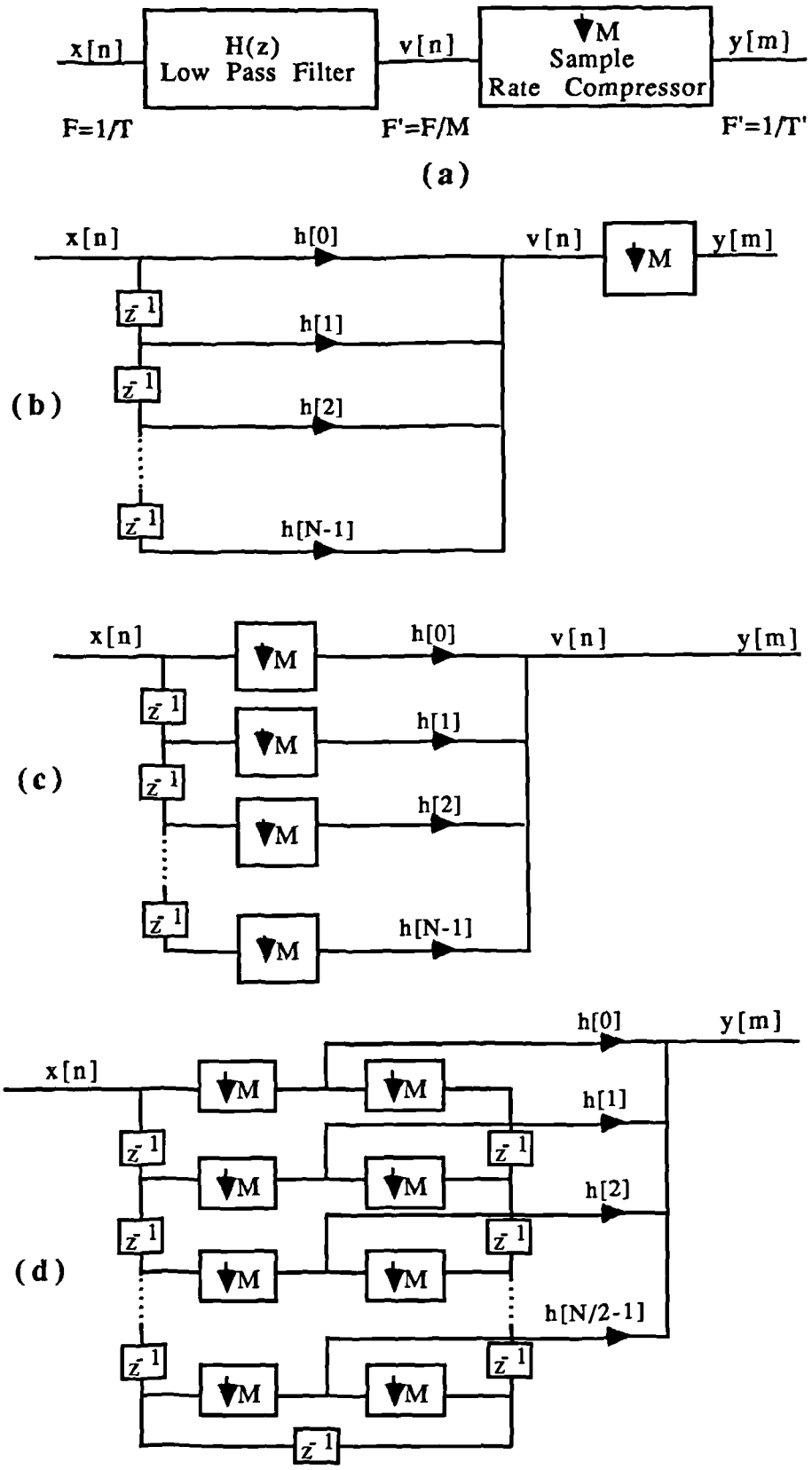


Fig. 3.8: Obtaining Efficient Decimator Structures (assuming N even for (d))

3.3.2 Multi-Stage Structures

Larger changes in sampling rate can be realized more efficiently by a cascade of independent sampling rate converters with smaller conversion factors. This is shown in Figs. 3.9a-b for I stage interpolation and decimation, respectively, with:

$$L = \prod_{i=0}^{I-1} L_i \tag{3.20}$$

and

$$M = \prod_{i=0}^{I-1} M_i \tag{3.21}$$

Increased efficiency is obtained by designing the low pass filters of some of the stages to have large transition bands (and hence low order). This is done in the knowledge that filters in other stages will ensure that overall aliasing (for decimation) or imaging (for interpolation) will be kept to within prescribed limits.

As a specific example, consider the task of raising the sampling rate of a signal by a factor of 16. Two possible structures are shown in Fig. 3.10a-b. The first is a single stage interpolator while the second is decomposed into two stages with $L_0 = 2$ and $L_1 = 8$.* Let the input sampling rate be $F = F_0 = 44.1kHz$, the standard digital audio rate. Also set the passband and stopband frequency edges to $f_p = 20.0kHz$ and $f_s = 24.1kHz$, respectively, with passband and stopband ripples specified at $\delta_p = 1\text{x}10^{-4}$ and $\delta_s = 1\text{x}10^{-5}$, respectively. The stopband frequency is such that a certain degree of imaging will take place over the frequency region $f \in [F_0/2=22.05kHz, f_s=24.1kHz]$.

In the single stage implementation the low pass filter will possess f_p, f_s, δ_p, and δ_s as above with a passband gain of $G=L=16$ and a filter sampling rate of $F_s=705.6kHz$. We have chosen to use equiripple filters as designed in [Mc79]. As such using the standard equiripple filter length estimation formula [Va87] we obtain the huge $N=911$ as an estimate for the length of the filter in this single stage implementation. This implies a computation rate of roughly:

$$R^{*}_{I=1} = \frac{N}{2L}F_s = \frac{911}{2\cdot 16}705600.0 \approx 20.1\text{x}10^6 multiplies\ per\ second\ (mps) \tag{3.22a}$$

$$R^{+}_{I=1} = \frac{N}{L}F_s = \frac{911}{16}705600.0 \approx 40.2\text{x}10^6 adds\ per\ second\ (aps) \tag{3.22b}$$

* It can be shown that it is more efficient to have interpolation stages with relatively low increases in sampling rate precede those with higher increases. The opposite is true for decimators (i.e., higher efficiency is obtained when large reductions in sampling rate are followed by smaller reductions).

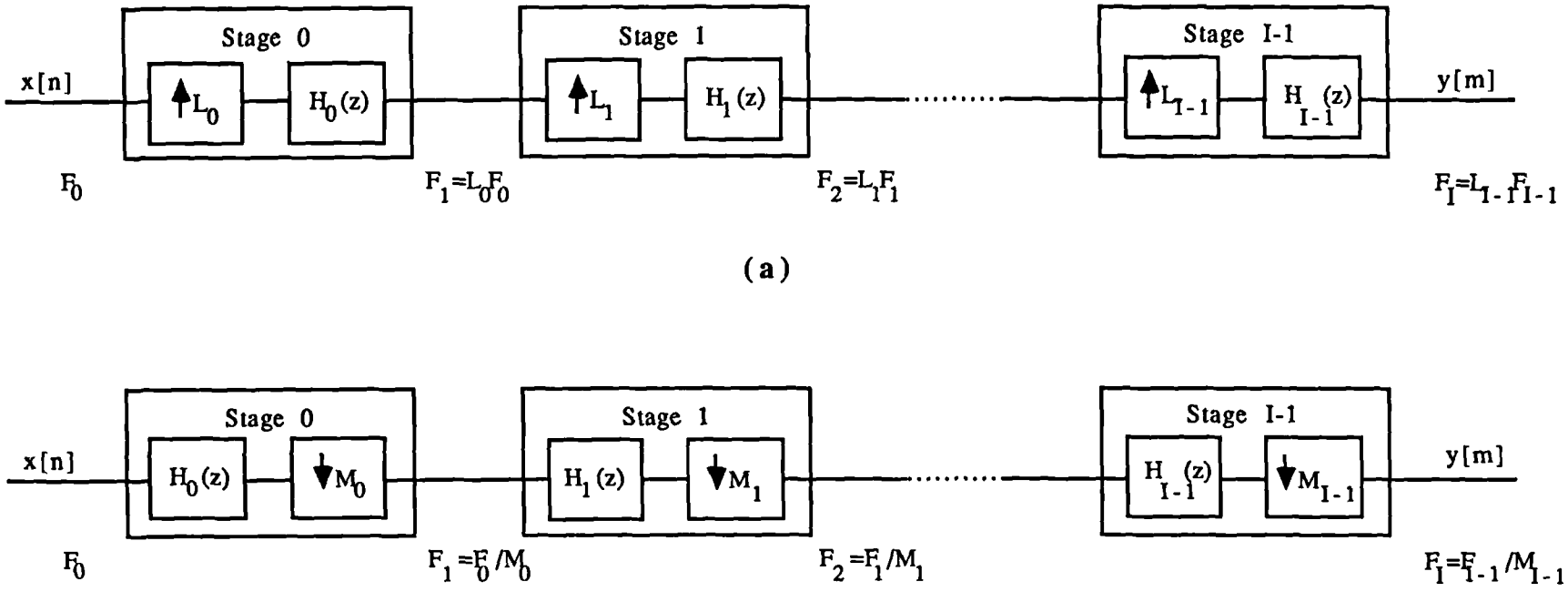
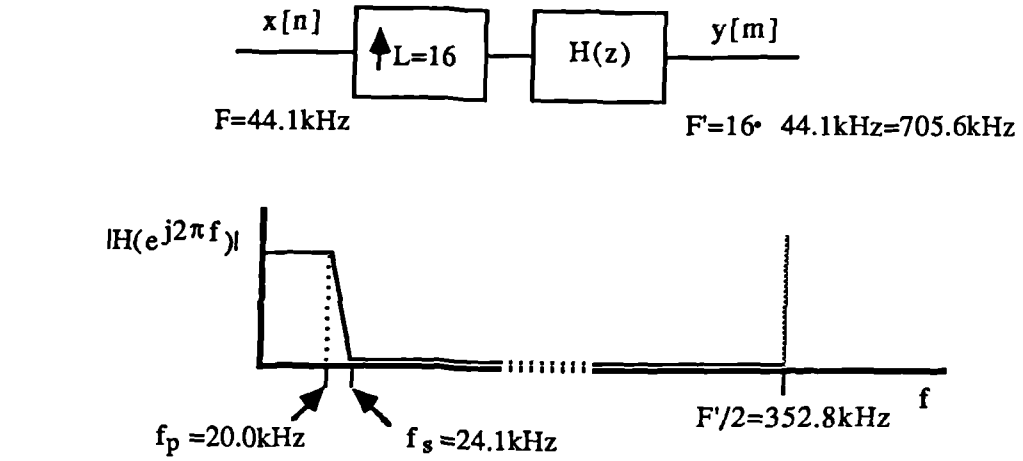
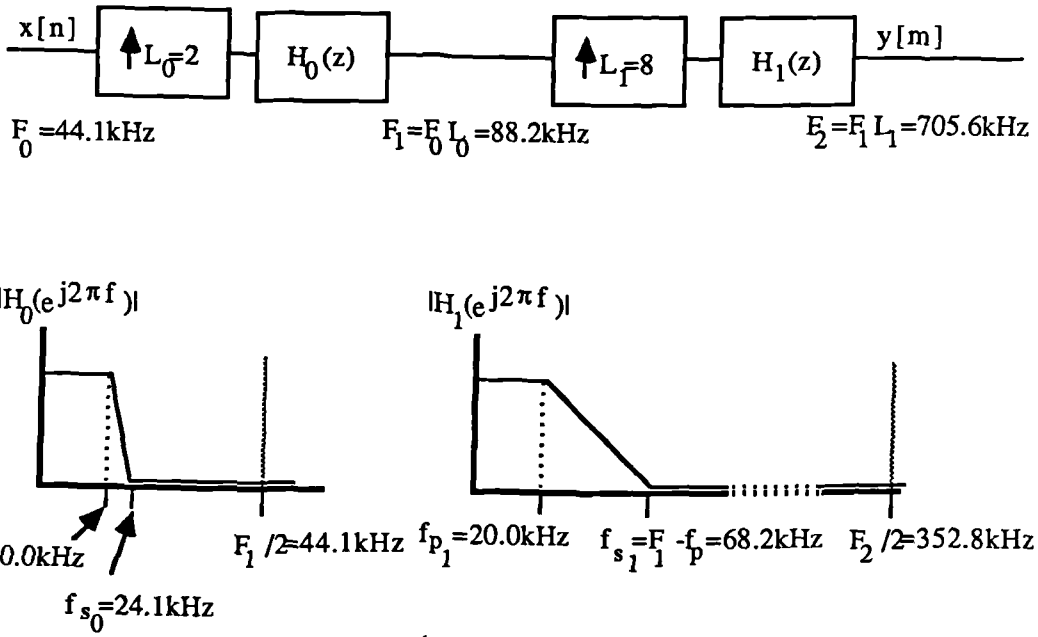


Fig. 3.9: Multi-Stage Interpolation and Decimation



(a)



(b)

Fig. 3.10: Examples of Single Stage and Multi-stage Interpolation

assuming the efficient implementation described in Fig. 3.7d. By contrast the design filter requirements for the two stage implementation are:

Stage 1:	$F_{s_0}=F_1=L_0F_0$	$2{\cdot}44.1kHz=88.2kHz$
	$f_{p_0}=f_p$	$20.0kHz$
	$f_{s_0}=f_s$	$24.1kHz$
	$\delta_{p_0}=\frac{\delta_p}{I}$	$\frac{1\text{x}10^{-4}}{2}=5\text{x}10^{-5}$
	$\delta_{s_0}=\delta_s$	$1\text{x}10^{-5}$
Stage 2:	$F_{s_1}=F_2=L_1F_1$	$8{\cdot}88.2kHz=705.6kHz$
	$f_{p_1}=f_p$	$20.0kHz$
	$f_{s_1}=F_1-f_p$	$88.2kHz-20.0kHz=68.2kHz$
	$\delta_{p_1}=\frac{\delta_p}{I}$	$\frac{1\text{x}10^{-4}}{2}=5\text{x}10^{-5}$
	$\delta_{s_1}=\delta_s$	$1\text{x}10^{-5}$

F_{s_i} is the sampling rate associated with the filter in the ith stage, which for design purposes is equal to the sampling at the output of the stage. f_{p_i} and f_{s_i} are the passband and stopband edges, respectively, for the filter used in the ith interpolation stage. δ_{p_i} and δ_{s_i} are the passband and stopband ripples, respectively, for the ith interpolation stage. It is important to note that in general for multi-stage implementations to satisfy the pass band ripple requirements it is necessary tighten δ_{p_i}, the ripple requirements on each stage:

$$\delta_{p_i} = \frac{\delta_p}{I} \qquad i \in \{0,1,\cdots,I-1\} \tag{3.23}$$

This ensures that additive ripple effects from each stage will not lead to a total pass band ripple greater than the original specification, δ_p. Also, the coefficients of the first stage filter and the second stage filter are to be scaled by $G_0=L_0=2$ and $G_1=L_1=8$, respectively, to achieve overall correct scaling. Estimates of the lengths of the two filters are given as $N_0 = 119$ and $N_1 = 81$. These imply a computation rate of:

$$R^*_{I=2} = \frac{N_0}{2{\cdot}L_0}F_1 + \frac{N_1}{2{\cdot}L_1}F_2 = \frac{119}{2{\cdot}2}88.2kHz + \frac{81}{2{\cdot}8}705.6kHz \approx 6.20\text{x}10^6 mps \tag{3.24a}$$

$$R^+_{I=2} = \frac{N_0}{L_0}F_1 + \frac{N_1}{L_1}F_2 = \frac{119}{2}88.2kHz + \frac{81}{8}705.6kHz \approx 12.4\text{x}10^6 aps \tag{3.24b}$$

We see that there is nearly a factor of four reduction in the computational complexity of the two stage implementation over the single stage implementation. For larger conversion

factors larger relative reductions in computational complexity are possible. Also, for large sample rate change factors further (but less dramatic) improvements in efficiency are often obtained by increasing the number of stages to three or four. Specific rules for choosing the number of stages as well as the conversion factor for each stage are given in [Cr83].

3.3.3 Special Classes of Filters for Efficient Implementations

Here we look at several classes of FIR filters which are particularly well suited to single stage and/or multi-stage sample rate conversion systems.

3.3.3.1 Halfband Filters

So-called "halfband" filters are symmetric about the quarter sampling rate, $\omega = \frac{\pi}{2}$ such that:

$$H(e^{j\omega}) = 1 - H(e^{j(\pi-\omega)}) \tag{3.25}$$

which implies that:

$$\delta_p = \delta_s \tag{3.26}$$

$$\omega_p = \pi - \omega_s \tag{3.27}$$

and

$$H(e^{j\pi/2}) = 0.5 \tag{3.28}$$

with δ_p and δ_s the passband and stopband ripples, respectively, and ω_p and ω_s the passband and stopband edge frequencies, respectively. The magnitude response of a low pass halfband filter is shown in Fig. 3.11a. It can also be shown that the impulse response of halfband FIR filters satisfy the following constraints:

$$h[n] = \begin{cases} 1 & n = 0 \\ 0 & n = 2k \quad k \in Z\ ,\ k \neq 0 \end{cases} \tag{3.29}$$

From the above constraints and the frequency response of Fig. 3.11a we see that halfband filters are particularly well suited to factor of two changes in sampling rate. The large number of zero-valued coefficients in the impulse response of these filters lower the computational complexity of a times two sample rate converter by about a factor of two over the efficient structures of the Section 3.3.1.

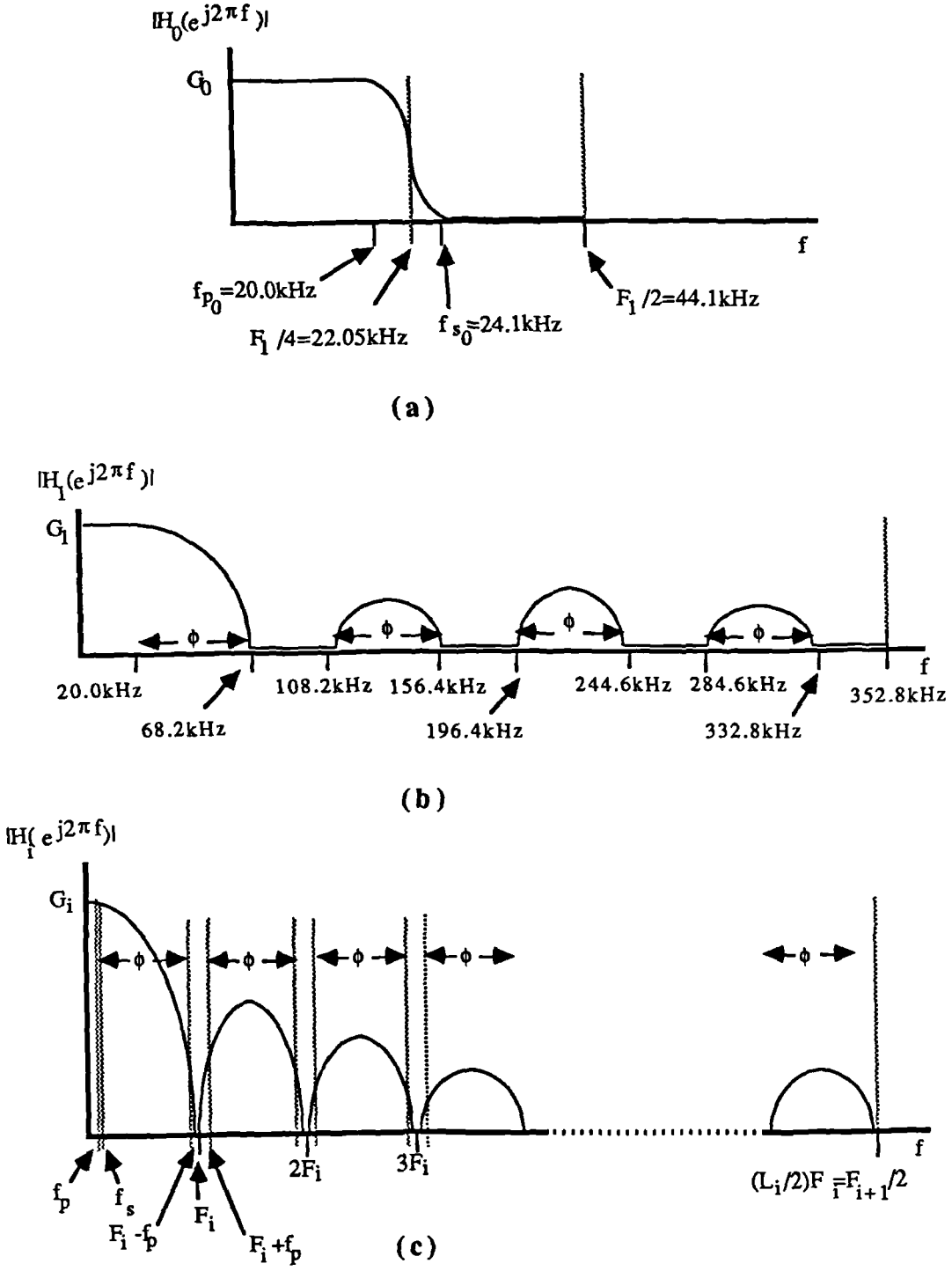


Fig. 3.11: Special Classes of Filters: (a) Halfband Filter, (b) Multiband Filter, (c) Comb Filter

In terms of the example in the previous sub-section, a halfband filter can be used in the first interpolation stage. The length of this filter with f_{p_0}=20.0kHz, f_{s_0}=24.1kHz, and $\delta_{p_0}=\delta_{s_0}$=1.0x10^{-5} is N_0=129. With the additional factor of two savings the total computation for the 16 times interpolator is now approximately:

$$R^{\bullet}_{I=2} \approx \frac{N_0}{2 \cdot 2L_0}F_1 + \frac{N_1}{2L_1}F_2 = \frac{129}{2 \cdot 2 \cdot 2}88.2kHz + \frac{81}{2 \cdot 8}705.6kHz \approx 4.99\text{x}10^6 mps \qquad (3.30a)$$

$$R^{+}_{I=2} \approx \frac{N_0}{2L_0}F_1 + \frac{N_1}{L_1}F_2 = \frac{129}{2 \cdot 2}88.2kHz + \frac{81}{8}705.6kHz \approx 9.99\text{x}10^6 aps \qquad (3.30a)$$

Another approach would be to cascade several times two sample rate conversion stages based on halfband filters. This can be a particularly efficient means of implementing power of two changes in sampling rate (i.e., $M = 2^k$ or $L = 2^k$ $k \in Z$, $k>0$).

3.3.3.2 Multi-band Filters

In multistage sample rate conversion structures we saw that low order filters with wide transition bands could be used to change the sampling rate of a signal by large factors more efficiently than in a single stage realization. The filters used in a multi-stage implementation are low pass filters with stopbands beginning at the low frequency edge of the lowest frequency spectral replicate to be attenuated by the filter and ending at half the sampling frequency. For example, the low pass filter in the second stage of the interpolator of Fig. 3.10b has an attenuation region of $f \in [F_1 - f_p$=68.2kHz,½F_2=352.8kHz). Additional savings in computation can be obtained if the filter is designed to attenuate only the required spectral replicates themselves and not the frequency bands between these replicates. For the second stage of the 16 times interpolator the replicates occupy the frequency regions: { (68.2kHz,108.2kHz), (156.4kHz,196.4kHz), (244.6kHz,284.6kHz), (332.8kHz,352.8kHz) }. We can design a filter which would attenuate the replicates but whose response would be left essentially unconstrained in the frequency regions between the replicates, the so-called "ϕ-bands." Unfortunately, no length estimation formula exists for such multi-band designs. Trial and error has shown that in our example the length could be reduced to N_1=75 with a response similar to that of Fig. 3.11b. This gives the somewhat smaller computation rate of:

$$R^{\bullet}_{I=2} \approx \frac{N_0}{2 \cdot 2L_0}F_1 + \frac{N_1}{2L_1}F_2 = \frac{129}{2 \cdot 2 \cdot 2}88.2kHz + \frac{75}{8 \cdot 2}705.6kHz \approx 4.73\text{x}10^6 mps \qquad (3.31a)$$

$$R^{+}_{I=2} \approx \frac{N_0}{2L_0}F_1 + \frac{N_1}{L_1}F_2 = \frac{129}{2 \cdot 2}88.2kHz + \frac{75}{8}705.6kHz \approx 9.46\text{x}10^6 aps \qquad (3.31b)$$

Larger reductions can be obtained if the ϕ-bands occupy a larger percentage of the total

band.

3.3.3.3 Comb Filters

Another related class of filters which are especially well suited to multi-stage cascade implementations is that of the comb filter. The impulse response of a comb filter of length N is:

$$h[n] = \begin{cases} 1 & n \in \{0,1,\cdots,N-1\} \\ 0 & n \notin \{0,1,\cdots,N-1\} \end{cases} \tag{3.32}$$

Its frequency response can be shown to be:

$$H(e^{j\omega}) = \frac{\sin(\frac{1}{2}\omega N)}{\sin(\frac{1}{2}\omega)} e^{-j\frac{1}{2}(N-1)\omega} \tag{3.33}$$

The magnitude response for $N=L_i$ is shown in Fig. 3.11c. The response is similar to that of the multi-band (i.e., ϕ-band) filters discussed earlier in this section with

$$N_i = L_i \tag{3.34a}$$

or

$$N_i = M_i \tag{3.34b}$$

Comb filters are particularly efficient because with all the coefficients equal to one no multiplications are necessary (except for a scale factor of $\frac{1}{M}$ for the decimator). Factor of M_i sample rate reduction with a comb filter of length, $N_i = M_i$, is achieved by partitioning the input signal into contiguous, M_i sample blocks and outputting the sum of the samples in each block. (For approximately unity passband filter gain it is necessary to scale the output sequence by $\frac{1}{M}$.) Factor of L_i interpolation with a comb filter of length, $N_i = L_i$, is achieved simply by repeating each input sample L_i times.

Comb filtering is most effective when the required attenuation regions centred at frequencies, $f = F_i, 2F_i, \cdots$, are of very small bandwidth compared to the input sampling rate, F_i. For an L_i times interpolator, raising the sampling rate to $F_{i+1} = L_i F_i$, the resulting passband and stopband ripples are given as:

$$\delta_s \geq \frac{|H(e^{j2\pi f_l})|}{|H(e^{j0})|} \geq \left| \frac{\sin(\pi f_s/F_i)}{L_i \sin[(\pi/L_i)(1-f_s/F_i)]} \right| \tag{3.35}$$

where

$$f_l = \frac{F_i - f_s}{F_{i+1}} \tag{3.36}$$

The smaller the attenuation region (i.e., the smaller the signal bandwidth), the larger the attenuation in that region. Greater levels of attenuation can be achieved by cascading more than one comb filter. When using a single comb filter the resulting pass band ripple is given as:

$$\frac{\delta_p}{I} \geq 1 - \frac{1}{L_i}\left|\frac{\sin(\pi f_p/F_i)}{\sin(\pi f_p/L_i F_i)}\right| \tag{3.37}$$

As the pass band increases so does pass band ripple.

3.4 Summary

In this chapter we have briefly surveyed the fundamental aspects of digital sample rate conversion techniques. We have seen how sample rate increase (interpolation) and sample rate decrease (decimation) are in essence complementary digital filtering operations. Various strategies for increasing the computational efficiency of these procedures have been described. These include the elimination of superfluous computations in the basic techniques, the use of multistage implementations, as well as the utilization of special classes of digital filters.

As mentioned before, in our application, decimation is used in the simulation software to help analyze the baseband performance of PWM DACs. Interpolation is used in the hardware implementation of a PWM based DAC to raise the input signal's sampling rate such that it is commensurate with the high pulse repetition frequency of the pulse width modulator. As such interpolation aids in the reduction of PWM distortion. In the next chapter we will see how interpolation in conjunction with noise shaping also helps make PWM DACs more practical to realize in hardware.

Chapter Four
Noise Shaping

4.1 Introduction

"Noise shaping" is a deterministic technique which uses a quantizer embedded in an error feedback loop to frequency shape a digital signal's quantization noise. A block diagram is shown in Fig. 4.1. The noise shaper accepts a high, b bit quality input and produces a low resolution, coarsely quantized b' bit output (i.e., $b>b'$). The quantizer coarsely *requantizes* the high resolution signal presented to its input by the limiter thus generating a requantization error. This error is frequency shaped by a filter and then fed back to the input. As we shall see, the noise can be attenuated over some portion of the total bandwidth at the expense of increasing it over other portions. This implies that noise shaping can be used to generate a coarsely quantized signal which accurately represents a finely quantized input over some limited portion of the total bandwidth. This further implies that if the noise shaper is preceded by an interpolator, requantization noise can be attenuated over the *entire baseband* of the original signal. Thus baseband signal quality can be retained with a smaller number of bits. It is this important property of ONS networks which make them attractive for use in PWM based DACs where, as we have seen in Chapter Two, excessive modulator clock speed (due to high signal wordlength) is a serious practical problem.

Noise shaping networks have their origins in a class of ADCs first introduced in [Cu60] and analyzed in [Sp62] in the early 1960's. With such converters it was possible to use oversampling and filtered error feedback techniques to achieve lower baseband noise power than with conventional ADCs. In effect, this is done by pushing the quantization noise outside the baseband where it can be removed subsequently by a digital filtering operation. From the mid-1970's similar techniques have been used with increasing success in higher accuracy noise shaping and Sigma Delta Modulation (SDM) based interpolative ADCs and DACs [Te90]. In general, these systems use some combination of

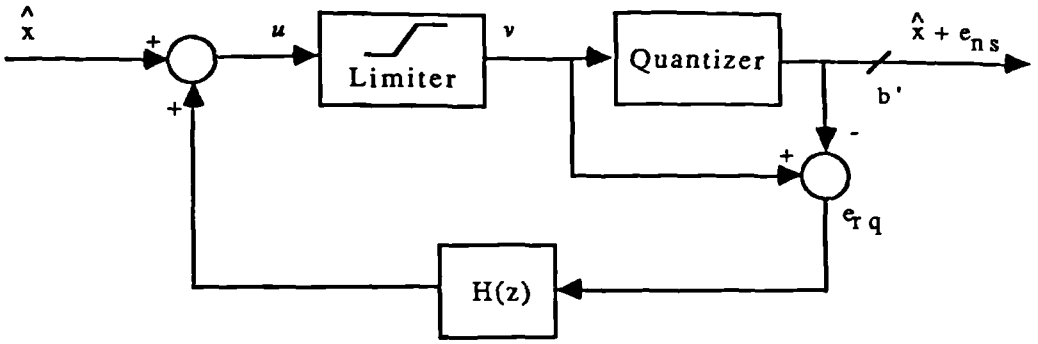


Fig. 4.1 Block Diagram of Noise Shaping Network

oversampling, feedback, and filtering with low resolution (often one bit) quantization to achieve high quality performance. This is done without the need for the high complexity, high precision analogue circuitry typically required in conventional converters.* Since these properties make interpolative converters ideal for VLSI implementation, they have become a popular alternative to the more traditional techniques. However, a disadvantage of these systems is that unwanted oscillations in higher (greater than second) order implementations often arise which can overload the quantizer and force the system to become unstable [Ca85, Ar87]. Consequently, new high order multi-bit and/or multistage converters have been designed to give high quality, highly stable performance [Ma87, Ma89, Ca89]. In addition, a number of recent theoretical [Ge89, Gr87] and practical [Ho91] results have helped to expand the general body of knowledge surrounding interpolative ADCs and DACs and have provided engineers with greater flexibility in particular design applications.

This chapter begins with a basic analysis of multi-bit ONS networks. We then explain how ONS techniques can be used to make PWM practical to realize in hardware. Next, we establish special design considerations for the feedback filters of ONS networks used in conjunction with PWM. In particular, we review two important theoretical results related to a design procedure used for such filters. Numerical examples illustrate the

* However, there of course remains a need for some high precision analogue circuitry. Specifically, for SDM based DACs, the "local" one bit DAC (i.e., the device which converts the one bit digital SDM output into a two level analogue signal) is often comprised of a switched capacitor circuit which must be very accurate. In fact, circuit problems with such local DACs have led some to abandon single bit SDM in favour of multi-bit SDM with a local PWM based DAC [Ma89].

improved performance over networks with conventional feedback filters. Lastly, we note the importance of dither in decorrelating quantization error from the input signal.

4.2 Basic Analysis

An approximate analysis of the noise shaping network of Fig 4.1 has been shown to be simple [Ge89]. We shall define $\hat{x}[n], n \in Z$ as the high resolution input of bandwidth, f_b, and sampling rate, f_s. In our application this signal has typically undergone some oversampling before being applied to the noise shaper The oversampling factor is $L \equiv \frac{f_s}{2f_b} > 1$. $\hat{x}[n]$ is comprised of $x[n]$, the unquantized input, and $e_q[n]$, a "small" error due to the initial quantization of the signal with additional errors introduced by subsequent signal processing operations (such as interpolation). $y[n]$ is the coarsely quantized output. $e_{rq}[n]$ is the "large" requantization error, and $e_{ns}[n]$ is the frequency shaped error at the output. $\hat{X}(z)$, $X(z)$, $E_q(z)$, $Y(z)$, $E_{rq}(z)$, $E_{ns}(z)$ are the respective z-transforms of the above. The filter in the feedback loop is denoted by $H(z)$ of order N. To implement the noise shaper, $H(z)$ must be such that its output is delayed by at least one sample. This corresponds to the requirement: $h[0]=0$. The limiter simply clips signals whose magnitude exceeds the range of the quantizer input. The limiting function, $L(\cdot)$, is given as:

$$v \equiv L(u) = \begin{cases} Q_{max}-1 & u > Q_{max}-1 \\ u & -Q_{max} \le u \le Q_{max}-1 \\ -Q_{max} & u < -Q_{max} \end{cases} \tag{4.1}$$

where u and v represent the input and output of the limiter, respectively. The limiter is linear over the "normal" operating range, $u \in [-Q_{max}, Q_{max}-1]$.

$y[n]$ can be expressed simply as the sum of the finely quantized input and the noise shaped error:

$$y[n] = \hat{x}[n] + e_{ns}[n] \tag{4.2}$$

Assuming that the limiter operates solely in its linear region, it is also apparent from Fig. 4.1 that in the z-domain we have

$$E_{rq}(z) = \hat{X}(z) + E_{rq}(z)H(z) - [\hat{X}(z) + E_{ns}(z)] \tag{4.3}$$

which can be manipulated into

$$G(z) = \frac{E_{ns}(z)}{E_{rq}(z)} = H(z) - 1 \tag{4.4}$$

$G(z)$ is known as the "noise transfer function" (NTF).

To simplify the analysis, at this point it is common to model the quantizer as an additive, independent noise source as shown in Fig. 4.2. This corresponds to assuming that the statistical properties of the error generated by the quantizer closely approximate those of a wide sense stationary random process which is independent of the input signal. Such an assumption is usually reasonable for multi-bit cases. (Further discussion can be found in [Be48, Ge78, Ha91].) From the theory of linear systems with stochastic inputs it is seen that $S_{ns}(\omega)$, the power spectral density (PSD) of the output error, is a frequency weighted version of $S_{rq}(\omega)$, the PSD of the requantization error:

$$S_{ns}(\omega) = |G(e^{j\omega})|^2 S_{rq}(\omega) \tag{4.5}$$

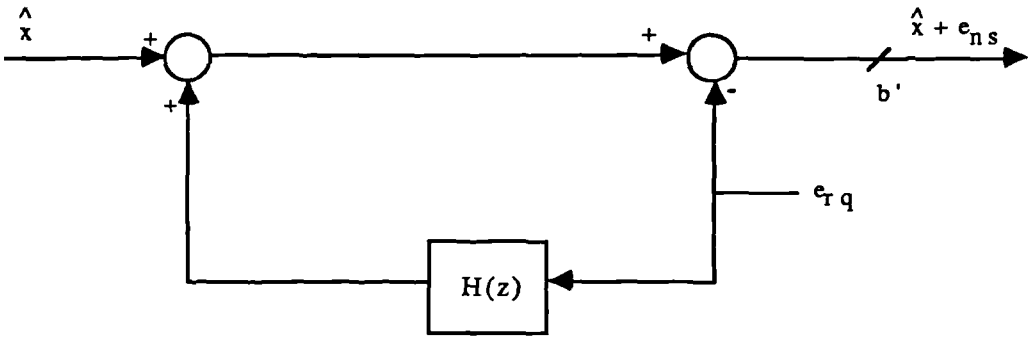


Fig. 4.2 Additive Noise Model for Noise Shaping Network

We can use any information we may have about the statistical properties exhibited by the requantization error signal to select an appropriate feedback filter, $H(z)$ (thus determining $G(z)$), to approximate some desired output error PSD. For example, if we assume that the error generated by the quantizer is white, then the PSD of the output error will have the same shape as $|G(e^{j\omega})|^2$. Even if this is not the case we can still compute the signal-to-noise ratio (SNR) over the baseband, $\omega \in [0,\pi/L]$:

$$SNR = \frac{P_x}{P_n} = \frac{\frac{1}{\pi}\int_0^{\pi/L} S_x(\omega)d\omega}{\frac{1}{\pi}\int_0^{\pi/L} [S_{ns}(\omega) + S_q(\omega)]d\omega} = \frac{P_x}{P_{ns_b} + P_q} \tag{4.6}$$

L is the oversampling ratio, P_x and P_n are the baseband power of the unquantized input

and the total output noise, respectively. $S_x(\omega)$ and $S_q(\omega)$ are the PSDs of the unquantized input and the small input quantization error, respectively. Lastly, P_{ns_b} and P_q are the baseband output error power and the original baseband input error power. Note that Eq. 4.6 assumes that $e_{ns}[n]$ and $e_q[n]$ are uncorrelated (which is reasonable).

Noise shapers cannot, of course, reduce the total output requantization error power associated with a regular b' bit linear PCM signal. Assuming that the requantization error is white, then again from the theory of linear systems with stochastic inputs, the ratio of the "input" signal power, P_{rq}, to the "output" signal power, P_{ns}, is given by the sum of the squares of the coefficients of the impulse response of the system, $G(z)$. That is, the total b' bit noise shaped error power (for $f \in [0, \frac{1}{2}f_s)$) exceeds the conventional b' bit non-noise shaped error power by a factor:

$$K = \frac{P_{ns}}{P_{rq}} = \sum_{n=0}^{N-1} g[n]^2 = 1.0 + \sum_{n=1}^{N-1} h[n]^2 \geq 1.0 \tag{4.7}$$

where we note from Eq. 4.4 that:

$$g[n] = \begin{cases} -1.0 & n=0 \\ h[n] & n \in \{1, 2, \cdots, N-1\} \\ 0.0 & n \text{ otherwise} \end{cases} \tag{4.8}$$

with $g[n]$ and $h[n]$ the impulse responses of the NTF and the Nth order FIR feedback filter, respectively. Equality for the far right-hand side of Eq. 4.7 (i.e., K=1.0) is attained only for the trivial case of cutting out the feedback filter (i.e., $h[n]$=0 for all n) resulting in conventional quantization. The noise power gain, K, can also be expressed in the frequency domain. Assuming $S_{rq}(\omega)$ is white (i.e., $S_{rq}(\omega)$ is a constant):

$$K = \frac{\frac{1}{\pi}\int_0^{\pi} S_{ns}(\omega)d\omega}{\frac{1}{\pi}\int_0^{\pi} S_{rq}(\omega)d\omega} = \frac{\frac{1}{\pi}\int_0^{\pi} |G(e^{j\omega})|^2 S_{rq}(\omega)d\omega}{\frac{1}{\pi}\int_0^{\pi} S_{rq}(\omega)d\omega} = \frac{1}{\pi}\int_0^{\pi} |G(e^{j\omega})|^2 d\omega \tag{4.9}$$

We see that the power gain can also be thought of as the average value of the squared magnitude frequency response of the NTF. This frequency domain expression can be obtained directly from Eq. 4.7 by Parseval's Theorem.

4.3 Noise Shaping For PWM Based DACs

In the context of PWM based DACs, noise shaping represents a means by which the wordlength of an oversampled signal can be considerably reduced with negligible loss in baseband signal quality. This decrease in wordlength implies a large reduction in the modulator clock speed. This reduction is essential for the hardware realization of a high quality, low cost DAC.

Recall from Chapter Two that oversampling is achieved by a procedure known as interpolation, which is essentially a digital filtering operation. If the noise shaper is preceded by an interpolator, the requantization error power can be redistributed over a larger bandwidth. Moreover, if $H(z)$ is chosen such that $|G(e^{j\omega})|^2$ has a high pass characteristic, then the low frequency baseband requantization noise power can be attenuated at the expense of increased high frequency noise power. (Remember that noise shaping cannot decrease the *total* noise power.) It is therefore possible to represent a high accuracy signal using fewer bits than typically necessary with virtually no loss in baseband signal quality. This approach is especially compatible with PWM which already requires some increase in pulse repetition frequency to reduce the types of distortion considered in Chapter Two. (A block diagram of a system using noise shaping and PWM is shown in Fig. 7.9 of Chapter Seven.)

In particular we see that a modulator with $\hat{b}$ bit input operating at a pulse repetition frequency of f_c must be able to produce pulses of $2^{\hat{b}}$ different widths. Recall from Chapter Two that this implies an internal modulator clock rate, f_{clk}, of at least:

$$f_{clk} = 2^{\hat{b}} f_c \tag{4.10}$$

for single sided (trailing edge) modulation or asymmetric double sided modulation.

$$f_{clk} = 2^{\hat{b}+1} f_c \tag{4.11}$$

for double sided symmetric or two sample consecutive modulation.* In a hardware implementation, f_{clk} will be somewhat larger due to the need for small guard bands in the PWM pulse interval to avoid interference from adjacent (non-ideal) pulses. (See Chapter Two.)

Table 4.1 shows modulator clock speed as a function of pulse repetition frequency, wordlength, and modulation type. Care must be taken in interpreting the clock rates shown in the table. We assume an original 16 bit signal sampled at $f_s = 44.1kHz$. With no oversampling, $f_c = 44.1kHz$, and the associated clock speed is near to 3GHz (for trailing edge modulation). Of course, the clock speed can be reduced by lowering the wordlength of the signal. However, with no oversampling, whether or not noise shaping is used, there will

* For the moment we restrict ourselves to UPWM modulation types.

Table 4.1: Modulator Clock Speed As a Function of Wordlength, Pulse Repetition Frequency, and Modulation Type								
f_c (kHz)	Modulation Type							
	wordlength (trailing edge & asymmetric)				wordlength (two sample consecutive & symmetric)			
	16	12	10	8	16	12	10	8
44.1	2.89GHz	181MHz	45.2MHz	11.3MHz	5.78GHz	361MHz	90.3MHz	22.6MHz
88.2	5.78GHz	361MHz	90.3MHz	22.6MHz	11.6GHz	723MHz	181MHz	45.2MHz
176.4	11.6GHz	723MHz	181MHz	45.2MHz	23.1GHz	1.45GHz	361MHz	90.3MHz
352.8	23.1GHz	1.45GHz	361MHz	90.3MHz	46.2GHz	2.89GHz	723MHz	181MHz

be a loss in signal quality. In any case, as we know from Chapter Two, oversampling is necessary to reduce PWM distortion. Small increases of the sampling rate to $f_s = 88.2kHz$ or $176.4kHz$ will not adequately reduce PWM distortion. In practice the situation is better when $f_s = f_c = 352.8kHz$. At this rate, if we noise shape down to eight bits, the modulator clock speed can be reduced to about 90.3MHz (orders of magnitude less than the 16 bit, 44.1kHz case).* This system has been shown to be practical for hardware implementation [Hi91].

4.4 NTF Design Considerations in ONS/PWM DACs

In this section we discuss how the feedback filter design considerations change for ONS networks used with PWM. Conventional designs concentrate on the satisfying baseband requirements and leave the high frequency response essentially unconstrained. By appealing to results presented later in this thesis we argue that NTFs designed for use in ONS/PWM DACs must also take the high frequency response into account.

* In principle, we can reduce the clock speed even further with more "aggressive" noise shaping. By attempting to decrease the wordlength to say four bits we increase requantization error power as well as output error power. Large increases in the former tend to result in large signals at the output of the feedback filter. This can overload the quantizer and produce distortion. Also, despite it being out-of-band, increased high frequency error power at the output of the noise shaper can create additional problems after modulation. This is discussed in the next section and more extensively in Chapter Seven.

4.4.1 The Problem with Popular Feedback Filters

A set of popular Nth order nonrecursive feedback filters has been derived in [Te78]. These filters satisfy the following relation:

$$G(z) = H(z) - 1 = -(1 - z^{-1})^N \tag{4.12}$$

They are robust in the sense that the filter coefficients have been derived independently of quantization error statistics and oversampling factor. For a given order, these NTFs achieve near to the lowest possible baseband output error power. They possess N zeros at $z=1$ and consequently have high pass frequency responses. The squared magnitude of the Nth order NTF is given as:

$$|G(e^{j\omega})|^2 = |1-e^{-j\omega}|^{2N} = [(1-\cos\omega)^2 + \sin^2\omega]^N = [2-2\cos\omega]^N \tag{4.13}$$

$$= [2\sin(\tfrac{1}{2}\omega)]^{2N}$$

Plots of third, fourth, and fifth order NTF magnitude responses are given in Figs. 4.3a-c, respectively. (Their impulse responses are listed in Appendix 4C.) In all cases $f_s = 352.8kHz$. The figure captions include K as computed by Eq. 4.7 and b', the output wordlength required to give roughly 16 bit baseband output quality. We notice that for a given quality specification more bits can be dropped as the order of the NTF increases. This is because baseband attenuation increases with N. We also see that K rises as the order is increased.

The filters of Eq. 4.12 have been used with great success in the past [Na87, Te78, Ma87, Ma89]. However, additional care must be taken when selecting a filter for use in an ONS/PWM DAC or digital power amplifier.

From Chapter Two we have seen that PWM is a nonlinear process which can produce baseband distortion for tone inputs. On this basis, it is plausible that if we noise shape prior to modulation, the spectrally shaped output noise can also be distorted by the modulation process. In fact, this will be shown to be the case in Chapter Seven where experimental results indicate that such distortion can manifest itself as increased *baseband* noise power. We will also see in Chapter Seven that this phenomenon is somewhat analogous to PWM foldback distortion for tone inputs. Recall that high frequency, high amplitude tones "fold back" from the carrier at multiples of the signal frequency. These terms can create large error components in the baseband when $|f_c + nf_v| < f_b$, particularly when there is little oversampling. In a similar manner, it is believed that high frequency noise generated by the noise shaper can be modulated back into the baseband. Loosely speaking, the larger the noise shaper's output noise power, the larger the noise foldback problem.

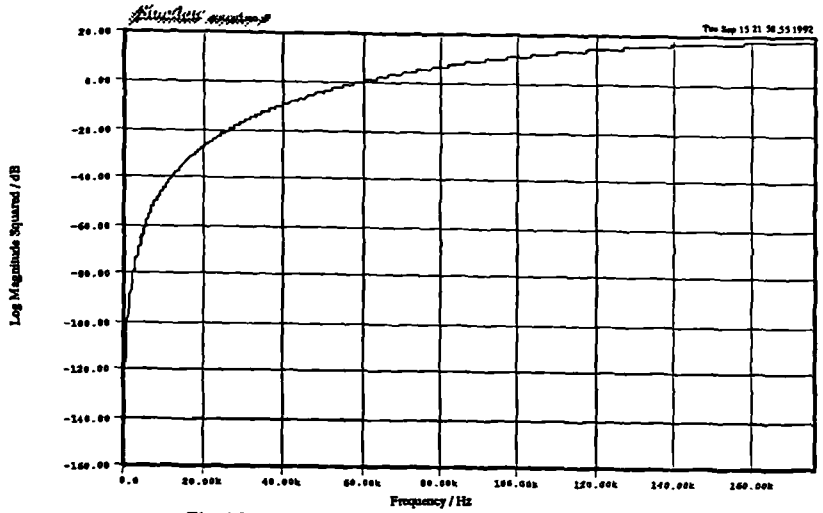


Fig. 4.3a: Conventional NTF Response (N=3 b'=12 K=20.0)

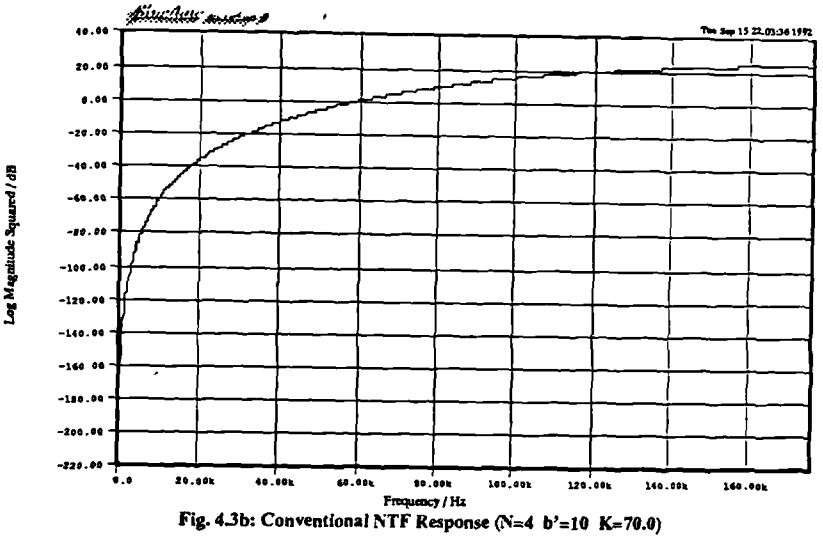


Fig. 4.3b: Conventional NTF Response (N=4 b'=10 K=70.0)

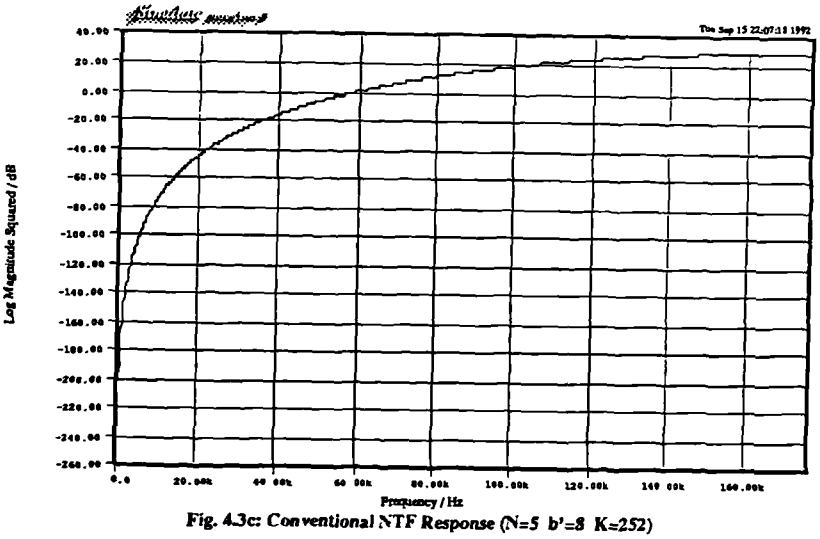


Fig. 4.3c: Conventional NTF Response (N=5 b'=8 K=252)

Therefore, these effects are often more prominent when high K NTFs are used.

While the NTF's of Eq. 4.12 have "optimal" baseband performance, they also tend to have high noise power gain at high frequencies (which tends to increase K). In Chapter Seven, we will see that the increased high frequency noise power from the noise shaping is large enough to cause undesirable baseband effects after modulation. The result is a severe loss in baseband SNR at the output of the modulator. In addition, high power gain corresponds to large coefficients in the feedback filter which gives rise to larger filter outputs. In general, these larger outputs will force the limiter into clipping more often than a low K filter. This clipping can produce baseband distortion of the signal at the output of the noise shaper.

4.4.2 New Feedback Filters

To address the problems discussed in the previous section, we use filters specifically designed for reducing both K, the total noise power gain before modulation, and the PWM noise modulation effect described above. The design of these filters is based on a numerical optimization procedure. Full details can be found in [Na91a, Hi92d].

In the present context, it is instructive to review two important theoretical results which form part of the basis of the filter design procedure. We begin with a preliminary definition and a theorem.

Definition:

> When $|G(e^{j\omega})|$, the magnitude of the frequency response of the NTF, is specified to within some positive, real, constant scale factor, then we say that the *shape of the NTF* has been specified.

Optimal Noise Shaping Theorem:

> If $G(z)$ is a stable, causal system of the form given by Eq. 4.4, then
>
> $$B = \int_0^{\pi} \log_2 |G(e^{j\omega})| d\omega \geq 0$$
>
> Equality ($B = 0$) is attained if and only if $G(z) = G_{mp}(z)$, the *minimum phase* realization (i.e., all poles and zeros inside the unit circle, $|z| \leq 1$) with the specified shape.

Under certain reasonable conditions B can be thought of as a measure of the loss of

information capacity of the noise shaping "channel" after noise shaping. Two formal proofs can be found in [Ge89]. However a more intuitive argument is also given. It is based on the fact that for a specified NTF shape, the information capacity of the noise shaped b' bit channel cannot exceed that of the non-noise shaped b' bit channel. This idea is shown more clearly in Fig. 4.4, the log magnitude frequency response of a typical NTF. Whenever the response dips below the 0dB line this represents a "local gain" in channel information capacity and whenever the response rises above the 0dB line this can be thought of as a "local loss" in channel information capacity. The above theorem says that α, the area of the shaded region below 0dB, must not exceed β, the area of the shaded region above 0dB. Since when $B=0$ the NTF must be minimum phase, $B>0$ corresponds to multiplying the magnitude of the frequency response of the minimum phase realization, $|G_{mp}(e^{j\omega})|$, by some scale factor greater than one. (Remember, the shape is independent of whether or not the NTF is minimum phase.) This has the effect of raising the total noise power gain, K, as defined in Eq. 4.7. Hence for a given NTF shape, the total noise power gain is minimized when the NTF is minimum phase and of the form given by Eq. 4.4.

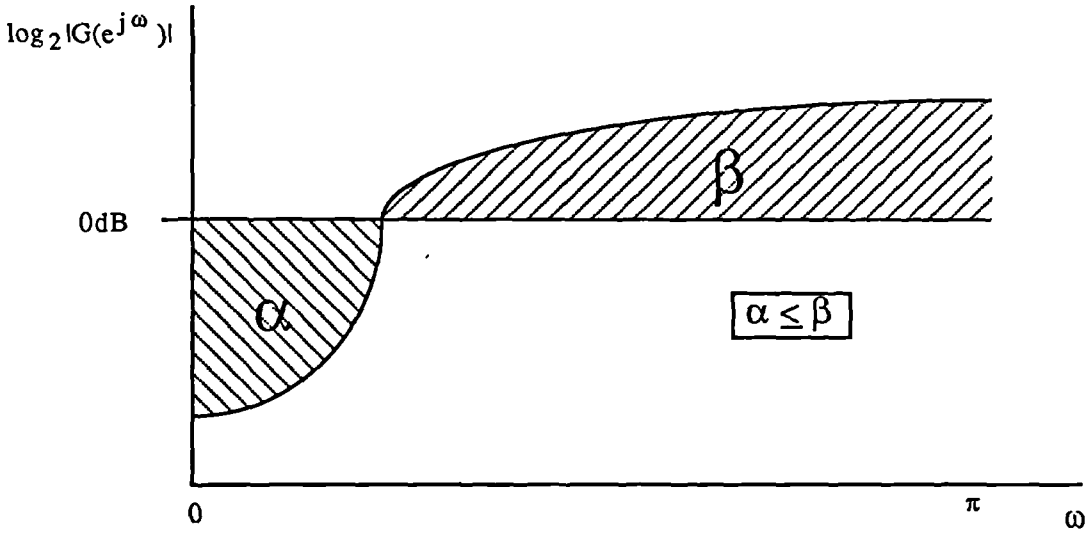


Fig. 4.4: Typical Magnitude Frequency Response for ONS NTF

From this theorem it can be seen that large high frequency gains of the NTFs shown in Fig. 4.3 are a consequence of the large baseband attenuation (i.e., large α implies large β). Given the fact that $e_q[n]$, the small initial error on the input signal, imposes a limit on the quality of the noise shaper output, there is no need for massive NTF attenuation of the requantization error. The reason the α region is so large for the standard NTF is because it

was in fact designed for *analogue-to-digital* conversion where with a "noise free" sampled analogue input, it is desirable to have the maximum baseband noise attenuation possible [Te78]. However, for DACs, where the digital input already has quantization noise inherently associated with it, there is no advantage in seeking such large baseband noise attenuation.

Assuming that all of the baseband is equally important, it makes more sense to specify $|G(e^{j\omega})|^2$, $\omega \in [0,\pi/L)$ to be at a *constant* level, G_α, giving the minimum amount of attenuation required. This will keep α small implying that it is at least possible for β (and consequently K) to be small. How do we determine G_α for a given output SNR requirement? For a b bit quality input signal and a b' bit resolution noise shaper output signal, the attenuation requirement is:

$$G_\alpha^2 = 2^{2(b'-1)} L \left[6 \cdot 10^{-SNR/10} - 2^{-2(b-1)} \right] \tag{4.14}$$

A derivation of this equation is given in Appendix 4B.

As it stands the Optimal Noise Shaping Theorem is useful only if the shape of $G(z)$ has been specified over the entire band, $\omega \in [0,\pi)$. When $|G(e^{j\omega})|$ is partially specified (such as over the signal band, $\omega \in [0,\pi/L)$) the question remains as to how to specify the shape in the remaining band(s) such that K is minimized. The answer is given by the following theorem:

Optimal NTF Shape Theorem:

> Given a specification for $|G(e^{j\omega})|$ over some portion of the total band, K will be minimized when the remainder of the band is specified as the single constant level which forces B to be zero.

In fact this theorem is still valid when more than one band is initially specified. To achieve minimum K, the magnitude of the NTF in remaining bands again should be specified to the constant level such that $B = 0$. The proof is given in [Na91b] and in Appendix 4A.

So, if we have specified the response over the baseband to be $|G(e^{j\omega})| = G_\alpha \;\; \omega \in [0,\pi/L]$, the above theorem tells us how to specify the remaining band (i.e, how to derive $G_\beta = |G(e^{j\omega})|$, $\omega \in [\pi/L,\pi)$ with $B=0$) such that K is minimized. In particular, under the minimum phase, $B=0$ condition, we see that:

$$\int_0^{1/2 f_s} \log_2 |G(e^{j2\pi f})| df = \int_0^{f_b} \log_2 G_\alpha df + \int_{f_b}^{1/2 f_s} \log_2 G_\beta df \tag{4.15}$$

$$= f_b \log_2 G_\alpha + (\tfrac{1}{2}f_s - f_b)\log_2 G_\beta = 0$$

Solving for G_β, we obtain:

$$G_\beta = 2^{\left[\frac{-f_b}{\frac{1}{2}f_s - f_b}\log_2 G_\alpha\right]} \tag{4.16}$$

The power gain can then be computed as:

$$K = \frac{2}{f_s}\int_0^{\frac{1}{2}f_s} |G(e^{j2\pi f})|^2 df = \frac{2}{f_s}\left[\int_0^{f_b} G_\alpha^2 df + \int_{f_b}^{\frac{1}{2}f_s} G_\beta^2 df\right] \tag{4.17}$$

$$= \frac{2}{f_s}\left[G_\alpha^2 f_b + G_\beta^2(\tfrac{1}{2}f_s - f_b)\right] \approx G_\beta^2\left[1 - \frac{1}{L}\right]$$

where the above approximation is valid for $G_\alpha^2 \ll G_\beta^2$.

So continuing with the b=16, b'=8, f_b=20kHz, f_s=352.8kHz example, what is the exact shape of the NTF which theoretically minimizes noise power gain? Let us specify the desired output SNR to be 97.5dB. We see that

$$L = \frac{f_s}{2f_b} = \frac{352.8kHz}{2 \cdot 20kHz} = 8.82 \tag{4.18}$$

Therefore, using Eq. 4.14, we have,

$$G_\alpha^2 = 2^{2(8-1)}8.82\left[6 \cdot 10^{-97.5/10} - 2^{-2(16-1)}\right] = 1.960\text{x}10^{-5} \quad (\text{or } -47.1\text{dB}) \tag{4.19}$$

This implies,

$$G_\beta = 2^{\left[\frac{-20}{\frac{1}{2}352.8-20}\log_2 1.960\text{x}10^{-5}\right]} = 2.0 \quad (\text{or } +6\text{dB}) \tag{4.20}$$

Thus the optimal shape is shown in Fig. 4.5, where for convenience we use $\log_{10}|G(e^{j\omega})|$ instead of $\log_2|G(e^{j\omega})|$. The power gain is given as:

$$K \approx 2^2\left[1 - \frac{1}{8.82}\right] = 3.55 \quad (\text{or } +5.50\text{dB}) \tag{4.21}$$

We stress that this is a *theoretical* lower bound and could not be realized with a practical, finite order NTF. Nevertheless, even low order approximations yield substantial improvements over the NTFs of Eq. 4.12.

As examples consider the plots shown in Fig. 4.6. They display the magnitude responses of the third, fourth, and fifth order optimized minimum phase NTFs which have been designed by the optimization procedure referred to above. (The coefficients are listed in Appendix 4C.) These NTFs are designed as alternatives to the three conventional NTFs

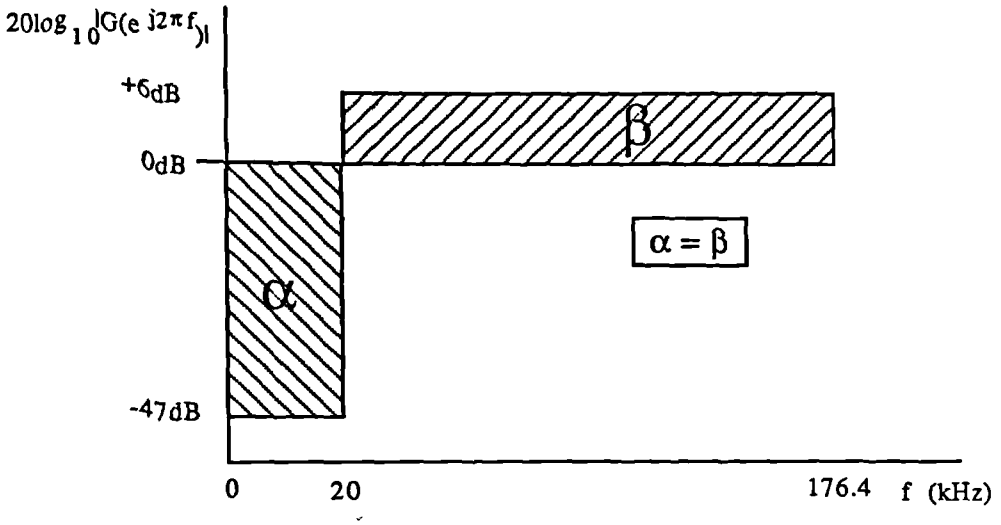


Fig. 4.5: Ideal (minimum K) NTF Magnitude Response

of Fig. 4.3. The noise power gains associated with these NTFs are compared with those of the corresponding conventional NTF in Table 4.2.

Table 4.2 Comparison of NTF Noise Power Gain

N	b'	K	
		Conventional	Optimized
3	12	20.0 (+13.0dB)	5.83 (+7.66dB)
4	10	70.0 (+18.5dB)	12.4 (+10.9dB)
5	8	252.0 (+24.0dB)	20.3 (+13.1dB)

Notice that in each case K is lower for the optimized NTFs. There is a particularly large reduction for the fifth order case. In Chapter Seven we will see that the use of optimized NTFs can lead to dramatic improvements in overall DAC performance.

The optimized NTFs introduced above have been used successfully with the "single sample per pulse" PWM modulation types. However, in Chapter Seven additional baseband noise problems are seen to arise when ONS networks are used with *two sample consecutive* PWM. Recall that the noise shaper output signal is made up of an input signal

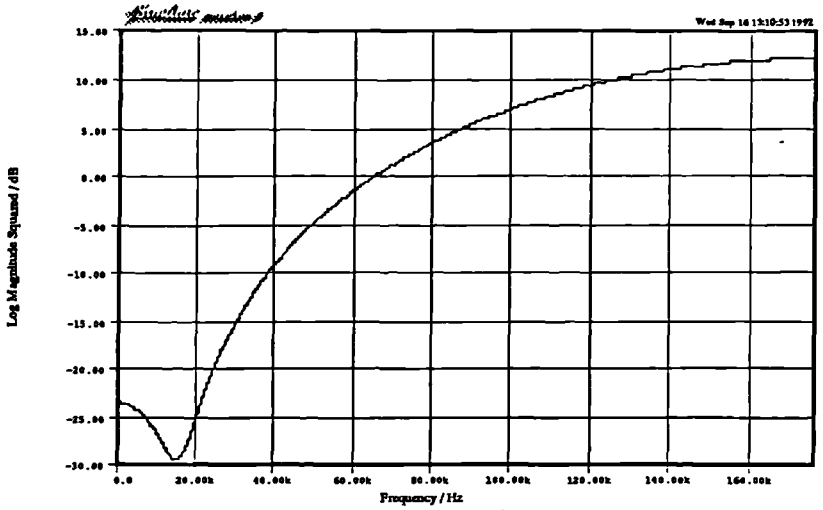


Fig. 4.6a: Optimized NTF Response (N=3 b'=12 K=5.83)

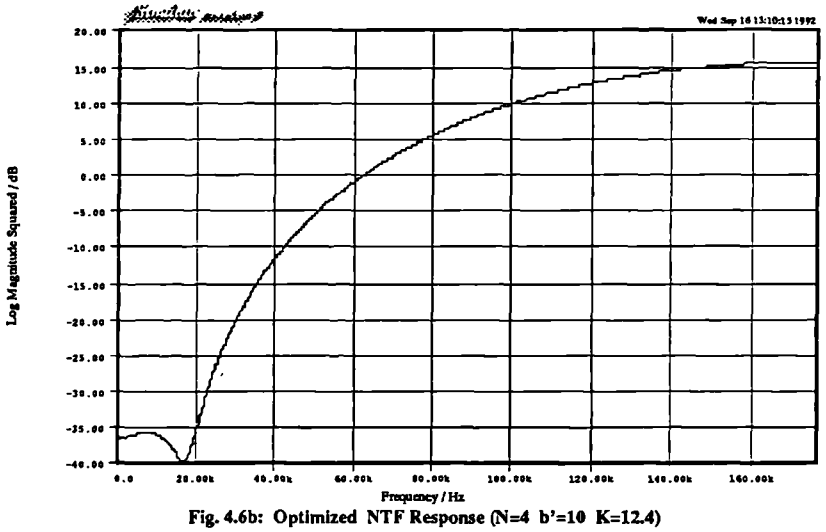


Fig. 4.6b: Optimized NTF Response (N=4 b'=10 K=12.4)

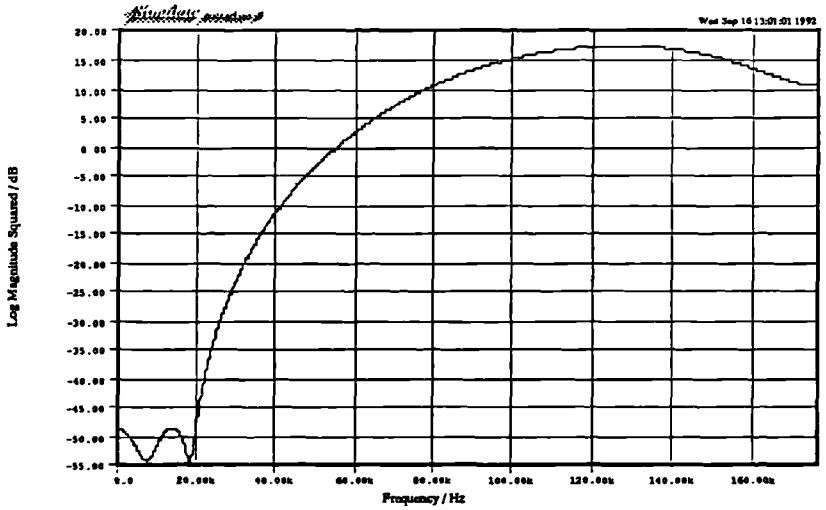


Fig. 4.6c: Optimized NTF Response (N=5 b'=8 K=20.2)

component, $\hat{x}[n]$, and an error signal component, $e_{ns}[n]$. These problems are believed to arise because the error components of the noise shaper output samples which vary the leading edges of the pulses are correlated with the error components of the samples that vary the trailing edges [Hi92e]. See Fig. 4.7. To combat this problem, we use a set of NTFs designed such that the "trailing edge" noise shaper error components are uncorrelated with the "leading edge" error components [Hi92ei, Pa92].

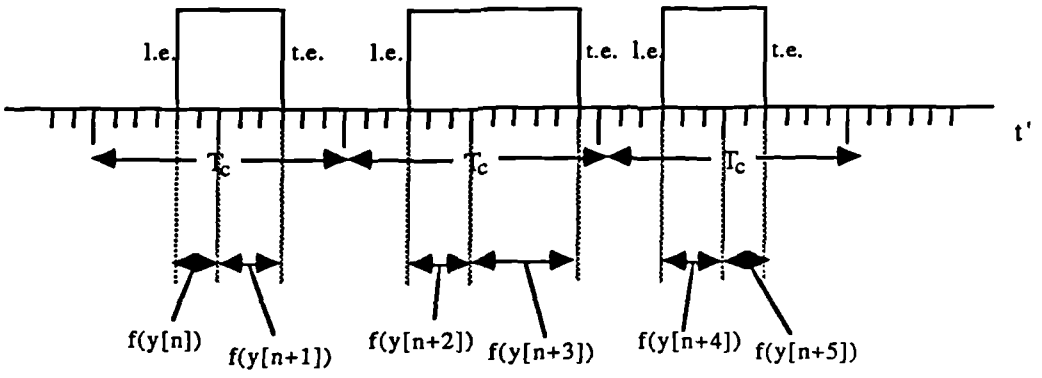


Fig. 4.7: ONS/UPWM Two Sample Consecutive Waveform (l.e.=leading edge t.e.=trailing edge) $y[i]=\hat{x}[i]+e_{ns}[i]$

In particular, recall that we model $e_{ns}[n]$ as the output of a linear system driven by an additive, independent noise source, $e_{rq}[n]$, which we regard as a stationary random process. The "output" autocorrelation sequence, $R_{ns}[m]$, is related to the "input" autocorrelation sequence, $R_{rq}[m]$, by [Pr88]:

$$R_{ns}[m] = E\left[e_{ns}[n]e_{ns}[n+m]\right] = \sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} g[k]g[j]R_{rq}[k-j+m] \tag{4.22}$$

where $E[\cdot]$ denotes expectation. When the requantization error is assumed white and of power, P_{rq}, then:

$$R_{rq}[m] = P_{rq}\delta[m] \tag{4.23}$$

where $\delta[m]$ is the unit sample sequence. This implies:

$$R_{ns}[m] = \sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} g[k]g[j]P_{rq}\delta[k-j+m] = P_{rq}\sum_{k=-\infty}^{\infty} g[k]g[k+m] \tag{4.24}$$

The requirement of having trailing edge noise shaping errors uncorrelated with leading edge noise shaping errors implies that:

$$R_{ns}[m] = 0 \quad \text{for all } m \text{ odd} \tag{4.25}$$

This condition is satisfied by a modified set of NTF impulse responses, $\hat{g}[k]$, with:

$$\hat{g}[k] = 0 \quad \text{for all } k \text{ odd} \quad (\hat{g}[0] = 1) \tag{4.26}$$

This corresponds to a zero-interleaved type NTF which, as we will see, creates symmetry in its magnitude response about $¼f_s$. Recall that for two sample consecutive UPWM, $f_s = 2f_c$. We assume $f_c = 352.8kHz$ which implies $f_s = 705.6kHz$. Adequate NTF baseband attenuation can be obtained when the NTFs of Figs. 4.3 and 4.6 (originally designed for $f_c = f_s = 352.8kHz$) are zero-interleaved and operated at the two sample consecutive signal sampling rate, $f_s = 705.6kHz$.* This is shown in Fig. 4.8 for the fifth order conventional and optimized NTFs. The new impulse response is expressed in terms of the old impulse response as:

$$\hat{g}[k] = \begin{cases} g[½k] & k \text{ even} \\ 0.0 & k \text{ odd} \end{cases} \tag{4.27}$$

In the frequency domain we have:

$$\hat{G}(e^{j\omega'}) = \sum_{k=-\infty}^{\infty} \hat{g}[k]e^{-j\omega' k} = \sum_{k=0,\pm2,\pm4\cdots}^{\pm\infty} \hat{g}[k]e^{-j\omega' k} = \sum_{m=-\infty}^{\infty} g[m]e^{-j2\omega' m} \tag{4.28}$$

$$= G(e^{j2\omega'}) = G(e^{j\omega})$$

where ω and $\omega'(=½\omega)$ are the normalized frequency variables for the single sample per pulse system and the two sample consecutive system, respectively. In Chapter Seven we will see how effective these modified NTFs are in eliminating the excess noise power problem associated with ONS/two sample consecutive UPWM DACs.

4.5 A Note on Dither

Although the use of dither has not been considered in this chapter we now briefly acknowledge its importance.

We have seen how effective ONS networks are in altering the frequency distribution of the requantization noise. However, it is sometimes possible for the quantizer to produce

* In fact, with the additional factor of two in the oversampling, these NTFs are overdesigned. From Eq. 4.14 we see that there is an additional three dB relaxation in the NTF baseband attenuation requirement.

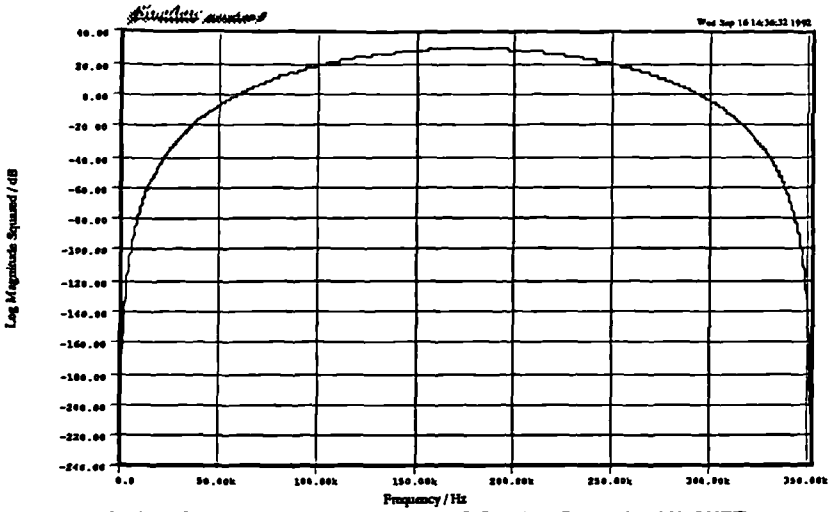


Fig. 4.8a: Conventional Zero-Interleaved NTF (Based on Conventional N=5 NTF)

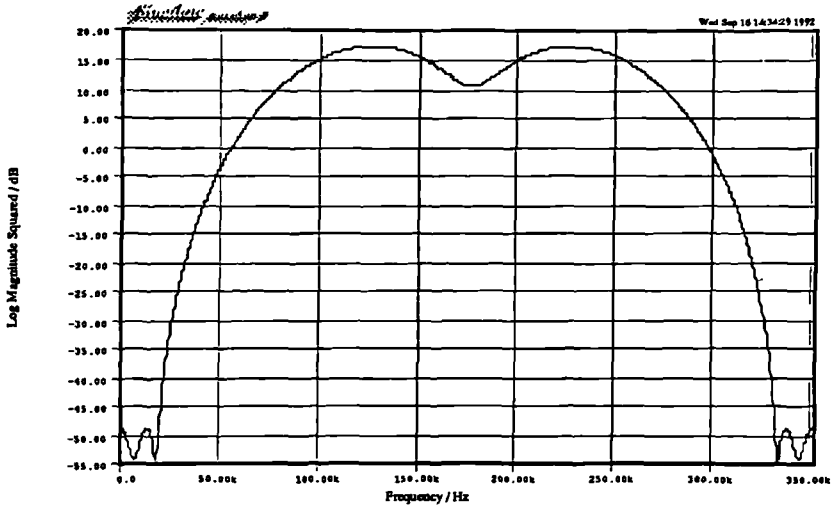


Fig. 4.8b: Optimized Zero-Interleaved NTF (Based on Optimized N=5 NTF)

requantization error which is not white but instead highly correlated with the signal. This has been shown to be the case for low level tone inputs where the requantization error was comprised of discrete frequency, limit cycle components [Va89]. Signal dependent requantization errors of this form are known to be subjectively undesirable in audio processing systems [Va84]. Of course, noise shaping will still attenuate these errors wherever the NTF is designed to attenuate requantization error. Nevertheless, in general, the highly correlated structure of such errors will remain. However, the addition of a small pseudo random "dither" sequence can be used to remove such correlations and whiten the requantization error associated with certain previously troublesome low level inputs [Va89].

A dithered noise shaper is shown in Fig. 4.9. Of course the addition of a such a noise source increases the total noise power. However, with the dither added directly before the quantizer as shown in the figure the dither sequence

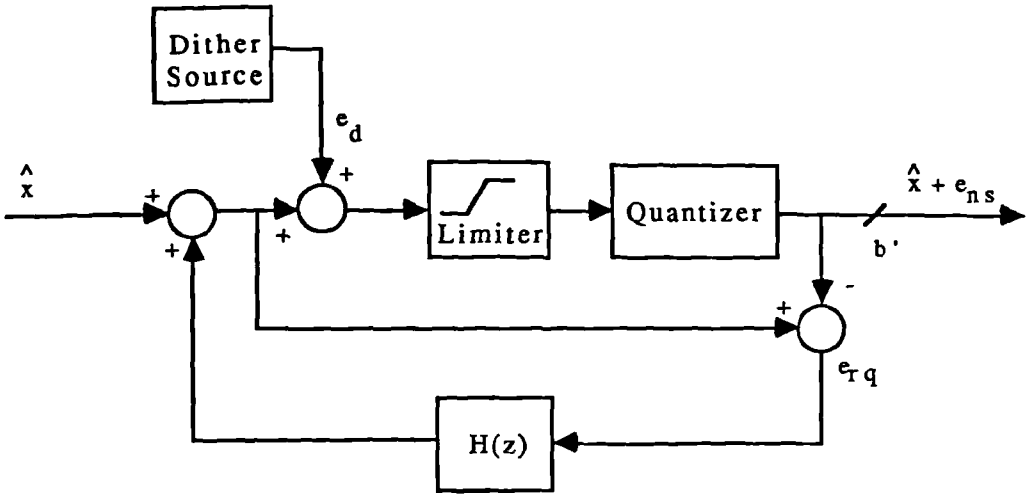


Fig. 4.9: Block Diagram of Noise Shaping Network with Dither

will also be noise shaped by the NTF such that the increase in baseband noise power will be negligibly small. The extent to which dither decorrelates the input signal from the requantization noise is a function of the statistical properties of the dither signal chosen. While the optimal choice of dither signal is still very much the subject of current research, white triangular probability density function (pdf) dither at $\pm\frac{1}{2}LSB$ of input signal resolution appears to give good results [Va89].

4.6 Summary

In this chapter the basic theory for ONS networks has been reviewed. The analysis is based on an additive, independent noise source model for the quantizer. In particular, expressions for the baseband output SNR as well as the total noise power gain for ONS networks have been given. It was shown how ONS helps in the design of *practical* high quality PWM DACs by allowing large reductions in modulator clock speed. Also, important issues in feedback filter design for ONS networks used with PWM have been presented. Attention has been focused on reducing noise power gain and on reducing an undesirable PWM noise effect. Two relevant theoretical results have been presented which

form part of the basis of an optimization procedure for designing feedback filters to be used in ONS/PWM systems. It has been shown that such filters can achieve significant reductions noise power gain. In Chapter Seven simulation results will show that these same filters can greatly reduce the PWM noise effect as well.

Appendix 4A

Proof of Optimal NTF Shape Theorem [Na91b]

To prove the Optimal NTF Shape Theorem we begin by considering two very small nonoverlapping, noncontiguous bands where the shape has not been specified. Moreover let these bands, $\omega \in [\omega_1, \omega_1+\delta]$ and $\omega \in [\omega_2, \omega_2+\delta]$, have equal width and be narrow enough so that $|G(e^{j\omega})|$ remains approximately constant at levels G_1 and G_2, respectively, over each band. This is shown in Fig. 4A.1. These levels are related by some positive constant γ:

$$\gamma \equiv \frac{G_2}{G_1} \tag{4A.1}$$

Next based on the Optimal Noise Shaping Theorem we define, B_Δ, an incremental information capacity loss function over these bands:

$$B_\Delta \equiv \int_{\omega_1}^{\omega_1+\delta} \log_2 G_1 d\omega + \int_{\omega_2}^{\omega_2+\delta} \log_2 G_2 d\omega = \delta[\log_2 G_1 + \log_2 G_2] = \delta \log_2 (G_1^2 \gamma) \tag{4A.2}$$

where the far right-hand equality follows from Eq. 4A.1. From Eq. 4.9 we also define an incremental noise power gain function:

$$K_\Delta \equiv \frac{1}{\pi}\left[\int_{\omega_1}^{\omega_1+\delta} G_1^2 d\omega + \int_{\omega_2}^{\omega_2+\delta} G_2^2 d\omega\right] = \frac{1}{\pi} G_1^2 \delta(1+\gamma^2) \tag{4A.3}$$

where G_2 is again eliminated by use of Eq. 4A.1.

Eq. 4A.2 can be solved for G_1:

$$G_1 = \gamma^{\frac{-1}{2}} 2^{\frac{B_\Delta}{2\delta}} \tag{4A.4}$$

Substituting Eq. 4A.3 into Eq. 4A.4 we obtain an expression for the incremental noise power gain in terms of γ:

$$K_\Delta = \frac{1}{\pi} 2^{\frac{B_\Delta}{\delta}} \delta\left[\gamma^{-1} + \gamma\right] \tag{4A.5}$$

Setting the partial derivative of Eq. 4A.5 with respect to γ to zero and solving for γ we find the value of γ which minimizes K_Δ:

$$\frac{\partial K_\Delta}{\partial \gamma} = \frac{1}{\pi} 2^{\frac{B_\Delta}{\delta}} \delta\left[-\gamma^{-2} + 1\right] = 0 \tag{4A.6}$$

The solution is $\gamma = 1$ with $\frac{\partial^2 K_\Delta}{\partial \gamma^2} > 0$ when $\gamma = 1$. This implies that when the output noise

levels, G_1 and G_2,in the two bands are equal, then the incremental noise power gain over the two bands is minimized. This result in fact generalizes to the whole of the band where the NTF shape is not specified. That is, the total noise power gain, K, can be minimized if $|G(e^{j\omega})|$ is set to a constant in the frequency bands where the shape was unspecified. (From the Optimal Noise Shaping Theorem this constant should be chosen such that $B=0$.) This is true because if we assume that a non constant shape can be found which minimizes K then it would be possible to find two narrow nonoverlapping, noncontiguous frequency bands with *different* noise levels which minimizes the incremental noise power gain function (thus making the total noise power gain smaller). But this is not possible as it would contradict Eq. 4A.6 for some ω_1 and ω_2 in the band where the shape was unspecified. In other words, if two separate frequency bands in the unspecified region with different magnitude response levels can be found, then K can be made smaller by making them the same level (i.e., by minimizing K_Δ). This implies that K is minimized when the whole of the unspecified band is a constant.

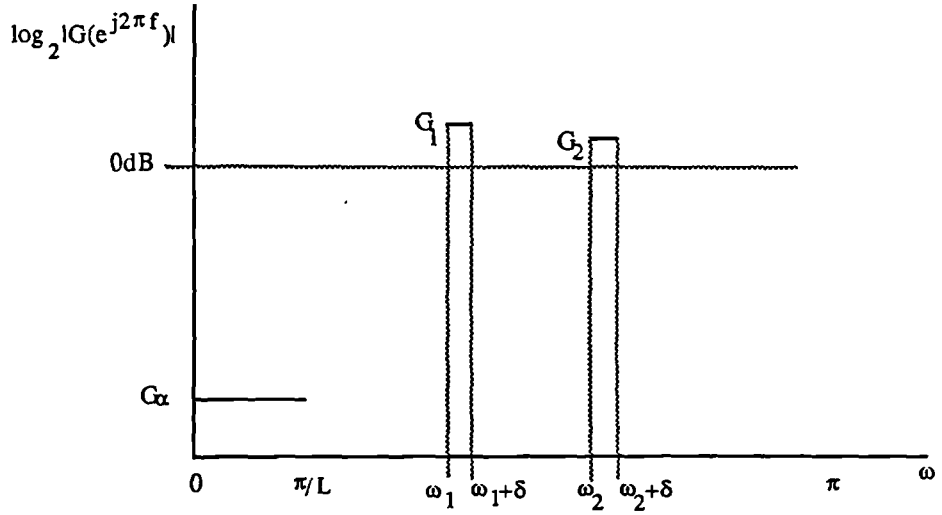


Fig. 4A.1: Optimal NTF Shape Theorem ($G_2 = G_1^\gamma$)

Appendix 4B

Derivation of Eq. 4.14

Consider a b bit quality input and a b' bit resolution noise shaper output. From Eq. 4.6 we have:

$$10\log\left[\frac{P_x}{P_{ns_b}+P_q}\right] = SNR\ (dB) \tag{4B.1}$$

or

$$P_x \cdot 10^{-SNR/10} = P_{ns_b} + P_q \tag{4B.2}$$

where P_x, P_q, and P_{ns_b} are the signal power, initial baseband quantization noise power (assumed white), and the shaped baseband requantization error power, respectively. Assuming a sinusoidal input signal with full scale magnitude, X_{max}, we have

$$P_x = \frac{X_{max}^2}{2} \tag{4B.3}$$

$$P_q = \frac{\left[\frac{X_{max}}{2^{b-1}}\right]^2}{12} \tag{4B.4}$$

$$P_{ns_b} = \frac{1}{\pi}\int_0^{\pi/L} S_{rq}(\omega)G_\alpha^2 d\omega = \frac{1}{\pi}\int_0^{\pi/L} P_{rq}G_\alpha^2 d\omega = \frac{1}{L}P_{rq}G_\alpha^2 \tag{4B.5}$$

Eqs. 4B.3 and 4B.4 are well known [Ja84]. The requantization error, P_{rq} (assumed white) is given by:

$$P_{rq} = \frac{\left[\frac{X_{max}}{2^{b'-1}}\right]^2}{12} \tag{4B.6}$$

where we assume that the maximum input to the quantizer is set to the maximum signal amplitude. From Eq. 4B.2 this implies that:

$$\frac{X_{max}^2}{2}10^{-SNR/10} - \frac{X_{max}^2}{12\cdot 2^{2(b-1)}} = \frac{X_{max}^2}{L\,12\cdot 2^{2(b'-1)}}G_\alpha^2 \tag{4B.7}$$

This can be re-expressed as:

$$G_\alpha^2 = 12\cdot 2^{2(b'-1)}L\left[\tfrac{1}{2}10^{-SNR/10} - \frac{1}{12\cdot 2^{2(b-1)}}\right] \tag{4B.8}$$

which simplifies to:

$$G_{\alpha}^{2} = 2^{2(b'-1)} L \left[6 \cdot 10^{-SNR/10} - 2^{-2(b-1)} \right] \qquad \text{(4B.9)}$$

as shown in Eq. 4.3.

Appendix 4C

Impulse Responses of Conventional and Optimized NTFs

Table 4C.1: NTFs with 12 Bit Output (N=3)		
Coefficients	Conventional	Optimized
$g[0]$	-1.0	-1.0
$g[1]$	3.0	1.9808268
$g[2]$	-3.0	-1.084262
$g[3]$	1.0	1.084943×10^{-1}
K	20.0	5.83

Table 4C.2: NTFs with 10 Bit Output (N=4)		
Coefficients	Conventional	Optimized
$g[0]$	-1.0	-1.0
$g[1]$	4.0	2.552653
$g[2]$	-6.0	-2.154382
$g[3]$	4.0	5.0011088
$g[4]$	-1.0	8.550248×10^{-2}
K	70.0	12.4

Table 4C.3: NTFs with 8 Bit Output (N=5)		
Coefficients	Conventional	Optimized
$g[0]$	-1.0	-1.0
$g[1]$	5.0	2.976059
$g[2]$	-10.0	-2.501200
$g[3]$	10.0	-6.206618×10^{-1}
$g[4]$	-5.0	1.812384
$g[5]$	1.0	-6.703392×10^{-1}
K	252.0	20.2

Chapter Five
Pseudo-Natural Pulse Width Modulation

5.1 Introduction

In Chapter Two we introduced the various PWM modulation types considered in this thesis. In particular, we examined two natural sampling PWM modulation types (single sided and double sided NPWM) and three uniform sampling PWM modulation types (single sided, double sided, and two sample consecutive UPWM). For tone modulation we saw that all five modulation types may exhibit so-called "foldback" distortion in the baseband (i.e., some of the sideband terms at multiples of the input tone frequency around the carrier and its harmonics can fall into the baseband). UPWM also gave rise to harmonic distortion of the input.

We noted that both the baseband harmonic and foldback distortion terms can be reduced by increasing ω_c, the pulse repetition frequency of the PWM waveform. In Section 2.5 of Chapter Two, we saw that for 16 bit quality only modest increases in ω_c were required to adequately suppress the baseband foldback distortion. However, for the three UPWM modulation types, excessively large increases in ω_c were necessary to reduce the *harmonic distortion* to levels beneath the noise floor. This is because UPWM baseband harmonic distortion decreases more slowly as a function of ω_c than UPWM and NPWM baseband foldback distortion. For our application, we would ideally prefer an *NPWM* based DAC which would give "distortion free" performance (i.e., all distortion well beneath the noise floor) at a reasonable pulse repetition frequency. In the past NPWM was thought to be unsuitable for a PWM based DAC as DAC inputs normally consist of *uniformly* spaced samples. (NPWM systems accept continuous time analogue inputs.) For this reason, in spite of poorer relative performance, the UPWM modulation types were viewed as the only sensible options for PWM based DACs [Sa83, Go91a, Ma89].

This chapter explores how we can achieve NPWM performance in a PWM based DAC. We begin by proposing a new technique which we call Pseudo-Natural Pulse Width

Modulation (PNPWM). PNPWM allows us to achieve the excellent performance associated with NPWM but in a fully digital, discrete time implementation. After introducing the basic idea, we present detailed descriptions of three possible implementations for a specific system. Detailed error analyses are given, and the issue of computational complexity is addressed. As future work we discuss possible alternative implementations which may be more computationally efficient.

5.2 The Basic Idea

How can we overcome the performance limitations imposed by the modulation processes normally considered for PWM based DACs? Recall from Chapter Three that the width of a PWM pulse can be specified by the time of intersection between $cw(t)$, a high frequency waveform of period, T_c, and $in(t)$, the analogue input signal (for NPWM), or $in(nT_s)$, a zero order sample-and-held version of this signal (for UPWM). This is shown in Fig. 5.1 for the case of single sided, trailing edge modulation ($T_c = T_s$). The sample values used to modulate the UPWM waveform are uniformly spaced in time. This is in contrast to NPWM where the samples used are irregularly spaced. The sampling instants (i.e., the times at which the samples are taken) are, in fact, signal dependent for NPWM.

The NPWM and UPWM pulse widths (and hence the NPWM and UPWM tone spectra) differ because the times at which $in(t)$ and $in(nT_s)$ intersect $cw(t)$ are different. This difference causes the presence (absence) of harmonic distortion in the UPWM (NPWM) case. The presence or absence of this harmonic distortion is a key factor in determining the overall level of performance obtainable in a PWM based DAC. Generally, the DAC input is a digital signal which corresponds, in some sense, to uniformly spaced samples of an analogue waveform. For this reason it was commonly believed that UPWM was the only choice for a PWM based DAC. (In the context of Fig. 5.1 there is an obvious correspondence between the uniformly spaced samples comprising the input to a DAC and the sample-and-held signal used in generating UPWM.) However, it has since been realized that this is an unnecessary restriction.

We propose to estimate for each PWM pulse the time at which $in(t)$ crosses $cw(t)$ and to apply this "cross point time" to an ordinary UPWM modulator to obtain approximately NPWM "distortion free" performance. This approximation to the cross point time will be based on nearby (uniformly spaced) input samples. We call this idea "Pseudo Natural Pulse Width Modulation" (PNPWM). A block diagram is shown in Fig. 5.2 where the "cross point deriver" is the box which converts the uniformly spaced sequence of input

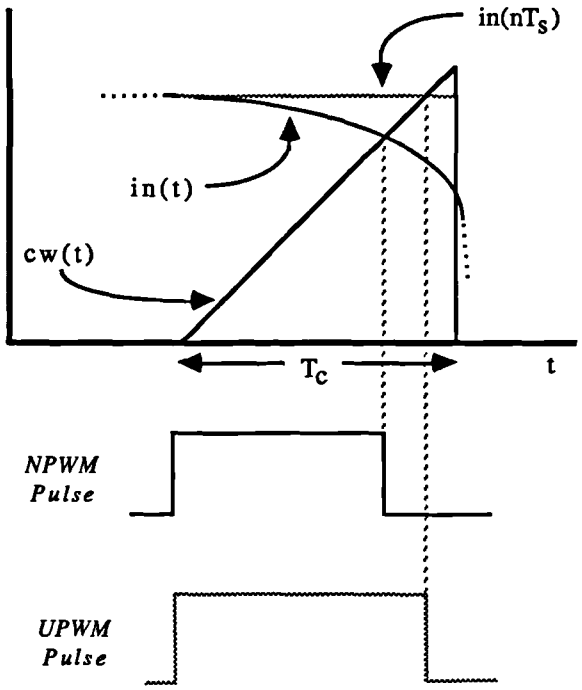


Fig: 5.1: Single Sided NPWM and UPWM Pulses ($T_s = T_c$)

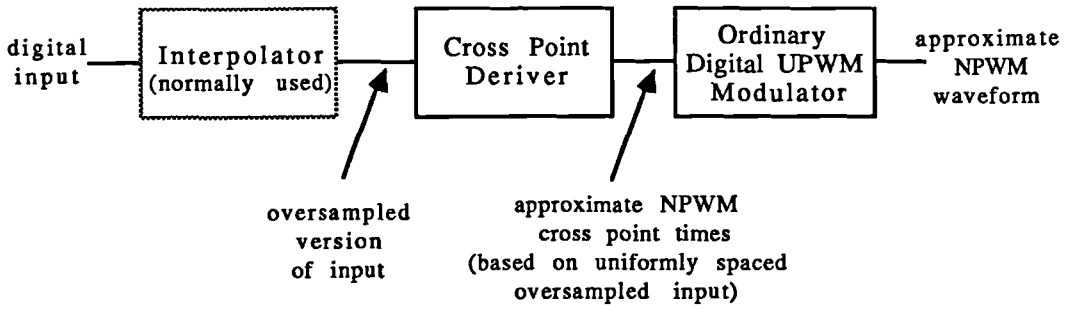


Fig. 5.2: A Basic PNPWM System (with interpolation)

samples to a sequence of approximate NPWM cross point times. (Thus, while being applied to the UPWM modulator at uniformly spaced instants of time, the cross point deriver output sequence consists of a nonuniform, signal dependent *resampling* of the input sequence.) The remainder of this chapter will focus on possible implementations of the cross point deriver.

5.3 Descriptions of Three Cross Point Algorithms

In this section we present thorough descriptions of three cross point computation procedures for PNPWM. In addition to a "functional" description of how each algorithm works, analyses of the computational complexity are also presented. Error analyses can be found in Appendices 5A, 5C, and 5D. The algorithms are designed for use in a standard digital audio application (i.e., a 16 bit, 20kHz bandwidth input signal sampled at 44.1kHz). We have chosen to consider single sided, trailing edge modulation over double sided modulation. While we saw in Section 2.5 that the lower baseband foldback distortion associated with the latter results in a slightly lower minimum ω_c for 16 bit quality (a possible advantage), we would have to compute *two* cross points per pulse. This can lead to a significant increase in the overall computational complexity of the algorithm (of course, a big disadvantage). Therefore, we concentrate on single sided PWM with the knowledge that the methods presented generalize to the double sided case in a straight forward way.

We have designed the algorithms to function within a DAC operating at a pulse repetition frequency eight times the sampling rate of the input (i.e., $L = 8, f_s = f_c = 8 \cdot 44.1kHz = 352.8kHz$). We do not claim that this choice is "optimal" in any sense. However, we do know from Section 2.5 that such a pulse repetition frequency is large enough to adequately suppress the single sided NPWM baseband foldback distortion. In addition, this choice of ω_c has been shown to be reasonable for a hardware implementation [Hi91].

The algorithms we now present are based on an intuitive "signal approximation" approach. Again we do not claim that these techniques are optimal—only that they can derive the cross point times with "reasonable" accuracy and with "reasonable" computation. The first algorithm is based on a first order polynomial approximation to the signal. As we shall see, for 16 bit quality applications, this is a crude but computationally efficient approach. The second and third algorithms use higher order approximations to the signal to obtain increasingly accurate estimates of the cross point times but with higher computational complexity.

5.3.1 Cross Point Computation Based on First Order Approximations

In this sub-section we introduce a crude first order cross point computation algorithm. After describing the procedure we address the issue of computational complexity. An error analysis for the technique is presented in Appendix 5A.

5.3.1.1 Description of Algorithm

This method is very simple. We form a straight line approximation to the signal and compute algebraically the time of the intersection between this approximation and the comparison waveform. This time may be used as an estimate to the true cross point time (i.e., the time when the comparison waveform and the actual analogue signal intersect). This is shown in Fig. 5.3. In the figure the comparison waveform is denoted as $cw(t)$ and can be described as a sawtooth function of period T_c where in each period:

$$cw(t) = t \qquad t \in [t_0, t_1) \tag{5.1}$$

$in(t)$ is the "underlying" analogue signal, samples of which comprise the digital input. In the figure we see two such adjacent samples, in_0 and in_1, at the uniformly spaced sampling instants, t_0 and t_1. They correspond in time to the beginning and end, respectively, of the period of the comparison waveform shown in the figure. The input signal and its samples are normalized to a maximum magnitude of 0.5 while t_0, t_1, and T_c are normalized to -0.5, 0.5, and 1.0, respectively. $\hat{in}_1(t)$, the straight line approximation to the signal, is formed by using the two adjacent input samples as shown below:

$$\hat{in}_1(t) = (in_1 - in_0)t + 0.5(in_1 + in_0) \tag{5.2}$$

Next, let us define an auxiliary function, $f(t)$:

$$f(t) \equiv cw(t) - in(t) \qquad f(t=t_\alpha) = 0 \tag{5.3}$$

The solution of the above equation, t_α, is the actual cross point time. In the context of our first order approximation, an estimate of the cross point time is given as the solution of a new equation, $\hat{f}_1(t) = 0$, where:

$$\hat{f}_1(t) \equiv cw(t) - \hat{in}_1(t) \qquad \hat{f}_1(t=t_2) = 0 \tag{5.4}$$

Hence, we have translated the cross point computation problem into a rootfinding problem. We denote the solution of Eq. 5.4 as t_2. Then from Eqs. 5.1 and 5.2 we have:

$$t_2 = \frac{0.5(in_1 + in_0)}{1 - in_1 + in_0} \tag{5.5}$$

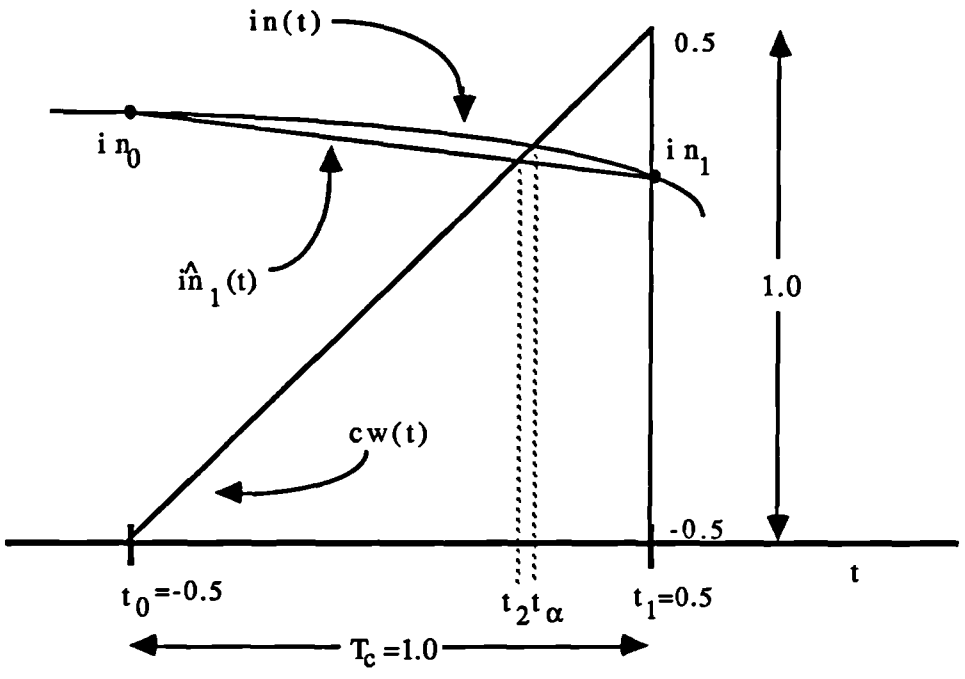


Fig. 5.3: First Order Approximation to the Cross Point Time

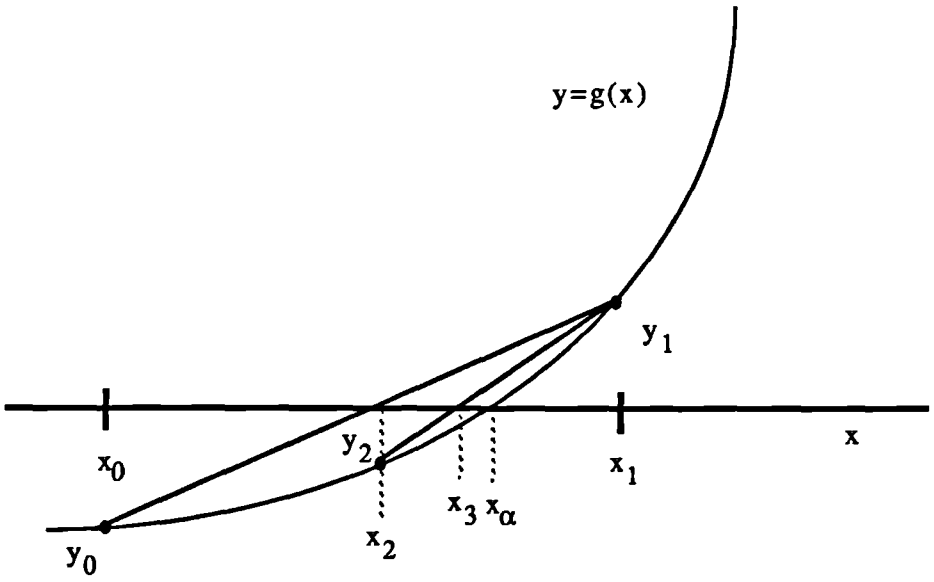


Fig. 5.4: The Secant Method

Due to the oversampling T_c is very small and $in(t)$ only exhibits gradual, smooth changes over $[t_0,t_1)$. Therefore, t_2 is a reasonable estimate of the true cross point time, t_α. It can be applied to a uniform sampling modulator to produce a crude version of natural sampling PWM. In fact, a broadly similar approach has been used in the double sided case [Me91].

The above expression for t_2 can be thought of as a "weighted average" of the input samples. This can be seen more clearly by rewriting Eq. 5.5:

$$t_2 = \frac{0.5(in_1 + in_0)}{1 - in_1 + in_0} = \frac{(0.5 - in_1)in_0 + (0.5 + in_0)in_1}{(0.5 - in_1) + (0.5 + in_0)} \tag{5.6}$$

$$= \frac{|f_1|}{|f_1| + |f_0|}in_0 + \frac{|f_0|}{|f_1| + |f_0|}in_1$$

where $f_0 = f(t_0)$ and $f_1 = f(t_1)$. So, for example, consider the case when $|f_1|$ is much smaller than $|f_0|$ (i.e., when the comparison waveform is nearer to the input signal at time t_1 than at t_0). We know that the signal exhibits small changes over $[t_0,t_1)$. Hence, the cross point time is near to t_1, and we weight in_1 more heavily than in_0 in our calculation of t_2.

An additional interpretation of this approach can be found among more general numerical techniques for approximating the solution of a nonlinear equation. In particular, Eq. 5.5 is equivalent to a single "secant iteration." The so-called "secant" method is an iterative numerical root finding procedure. It is based on forming straight line approximations to a function in the vicinity of one of the function's roots. An improved estimate of the function's root(s) is obtained by computing the root of the straight line approximation. This improved estimate can be used to form another straight line approximation, the root of which is an even better estimate. Convergence to the actual solution is guaranteed for "reasonable" functions with "reasonable" initial guesses to the root. (These conditions are easily satisfied in our application. See [At89] or [Sc89] for details.)

The technique is illustrated in Fig. 5.4 where we are trying to find the root, x_α, of a continuous, real valued function $y = g(x)$. x_0 and x_1 are taken as two initial guesses at x_α. We approximate $g(x)$ by the straight line specified by (x_0,y_0) and (x_1,y_1) and given as:

$$\frac{y_1 - y_0}{x_1 - x_0}(x - x_1) + y_1 \tag{5.7}$$

We set the above to zero and solve for x, calling the result x_2.

$$x_2 = x_1 - y_1\frac{x_1 - x_0}{y_1 - y_0} \tag{5.8}$$

This is used as an improved estimate of x_α. If we wish to obtain more accurate estimates of x_α we can use (x_1,y_1) and (x_2,y_2) in a straight line approximation and repeat the

procedure to generate $x_3, x_4, \cdots$ etc. The technique will produce iterates which converge to x_α when x_0 and x_1 are sufficiently close to x_α. The method generalizes as:

$$x_{j+1} = x_j - y_j \frac{x_j - x_{j-1}}{y_j - y_{j-1}} \qquad j \in Z \quad j>0 \tag{5.9}$$

In the case of our single iteration simple algebra shows that Eq. 5.5 is numerically equivalent to Eq. 5.8 with $x_j=t_j$, $j=1$, and $g(\cdot)=f(\cdot)$.

An error analysis for this technique is given in Appendix 5A. There we see that the secant method interpretation of our approach establishes a framework in which the error analysis for this technique is straightforward. Next, we examine the computational complexity of the algorithm.

5.3.1.2 Computational Complexity

A big advantage of this algorithm is its low computational complexity. The computation can be broken down as shown in Table 5.1.

Table 5.1: Computational Complexity of First Order Algorithm

Operation	Computation
$[in_0]+[in_1]$	1 add
$[0.5][(in_0+in_1)]$	1 multiply
$[1]-[in_1]+[in_0]$	2 adds
$\frac{[1]}{[(1-in_1+in_0)]}$	1 inversion
$[0.5(in_0+in_1)]\left[\frac{1}{(1-in_1+in_0)}\right]$	1 multiply
Total	3 adds, 2 multiplies, 1 inversion

For each entry in the table the operation considered is between two or more operands indicated by square brackets, [ ]. The majority of the computation will take place in performing the inversion. For example, on the Motorola 96000 DSP chip, a floating point inversion requires seven machine cycles (which is to be compared with a single machine cycle for a floating point multiply) [DS89]. The inversion uses a Newton-Raphson procedure which is based on approximations to the solution of equations such as:

$$h(x) = a - \frac{1}{x} \tag{5.10}$$

where a is the quantity to be inverted. (In our application $a = 1 - in_1 + in_0$.) The Newton-Raphson iterates would then take the form of:

$$x_n = x_{n-1}(2 - ax_{n-1}) \qquad n \in Z \quad n > 0 \tag{5.11}$$

In practice, the procedure would be terminated after a small number of iterations. An alternative approximation suggested by [Cr90] is based on the power series expansion:

$$\frac{1}{1-\delta} = 1 + \delta + \delta^2 + \delta^3 \cdots \qquad \delta < 1 \tag{5.12}$$

where, in our case, $\delta = in_1 - in_0$. An approximation to $\frac{1}{1-\delta}$ is obtained by truncating the series at δ^M:

$$\frac{1}{1-\delta} \approx 1 + \delta + \delta^2 + \cdots + \delta^M \tag{5.13}$$

This can be computed efficiently using nested multiplies: $1 + \delta\left[1 + \delta\left[1 + \delta\left[\cdots\right]\right]\right]$. It is easily seen that truncation to δ^M results in an error of $\frac{\delta^{M+1}}{1-\delta}$. However, at least in the case of the 96000 chip, it is more efficient to use the inversion routine which is already provided [Bo92].

5.3.2 Cross Point Computation Based on Fifth Order Approximations

In this section we describe a second, more accurate cross point computation time procedure. As before the technique is based on a "signal approximation/root finding" approach.

Recall that in the first algorithm we formed a straight line (i.e., first order) approximation to the analogue signal and algebraically computed the time of intersection between the approximation and the comparison waveform. While this method is computationally efficient, it does not yield even close to 16 bit quality results (see Appendix 5A and Chapter Seven.) To improve on this estimate, the second algorithm uses the result of the first algorithm as an "initial guess" to the cross point time to yield a much more accurate approximation. As before, we begin by describing the procedure and continue with an examination of the computational complexity. Relevant preliminary information on polynomial interpolation is contained in Appendix 5B, and an error analysis for this technique

is presented in Appendix 5C.

5.3.2.1 Description of Algorithm

Recall from Section 5.3.1 that the cross point time computation problem can be posed as the root finding problem: solve $f(t) = cw(t) - in(t) = 0$. The first algorithm effectively uses a secant iteration to derive t_2, an estimate to t_α, the actual solution of $f(t) = 0$ (i.e., the actual cross point time). Essentially, the second algorithm consists of a single Newton-Raphson iteration on $f(t)$ at t_2 to derive t_3, a more accurate estimate of t_α.

Before describing the specifics of the algorithm we comment briefly on the Newton-Raphson procedure as a general technique for estimating the solutions of nonlinear equations. In particular, consider a general real valued, differentiable function, $g(x)$, which passes through zero at least once over the interval of interest. As with the secant method discussed in Section 5.3.1, the Newton Raphson method is an iterative technique, producing a sequence of iterates, x_i, $i \in Z\ i \geq 0$, which, under reasonable conditions, converges to an actual true solution, x_α (i.e., $g(x_\alpha) = 0$). The ith iterate, x_i, is derived from the intersection between the x-axis and a straight line approximation of $g(x)$ taken at the previous iterate, x_{i-1}. Specifically,

$$x_i = x_{i-1} - \frac{g(x_{i-1})}{g'(x_{i-1})} \tag{5.14}$$

This procedure is shown in Fig. 5.5. The Newton-Raphson method produces iterates which converge to x_α more quickly than with the secant method. This may be attributed to the use of the derivative of the function. In fact, the secant method can be thought of as an approximate form of the Newton Raphson method where $\frac{x_{i-1} - x_{i-2}}{y_{i-1} - y_{i-2}}$ of Eq. 5.9 is taken as an estimate of $\frac{1}{g'(x_{i-1})}$ in Eq. 5.14.

In our case $f(t) = cw(t) - in(t)$ corresponds to $g(x)$. We perform a single Newton-Raphson iteration on $f(t)$ at $t=t_2$, the estimate to the cross point time produced by the secant iteration described in Section 5.3.1. This yields t_3, a near to 16 bit quality estimate of t_α. (See Appendix 5C.):

$$t_\alpha \approx t_3 = t_2 - \frac{f(t_2)}{f'(t_2)} = t_2 - \frac{cw(t_2) - in(t_2)}{cw'(t_2) - in'(t_2)} = t_2 - \frac{t_2 - in(t_2)}{1 - in'(t_2)} \tag{5.15}$$

It is this value, t_3, (or some quantity related to it) that can be sent to the UPWM modulator to obtain near to NPWM performance.

However, as $in(t_2)$ and $in'(t_2)$ are not explicitly available we form *approximations* to the input signal and its derivative. In particular, we use fifth order interpolation

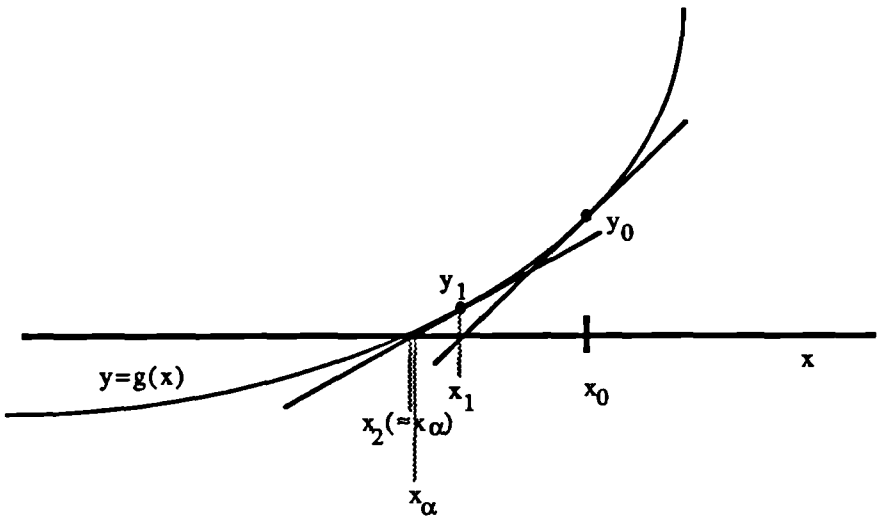


Fig. 5.5: The Newton-Raphson Method

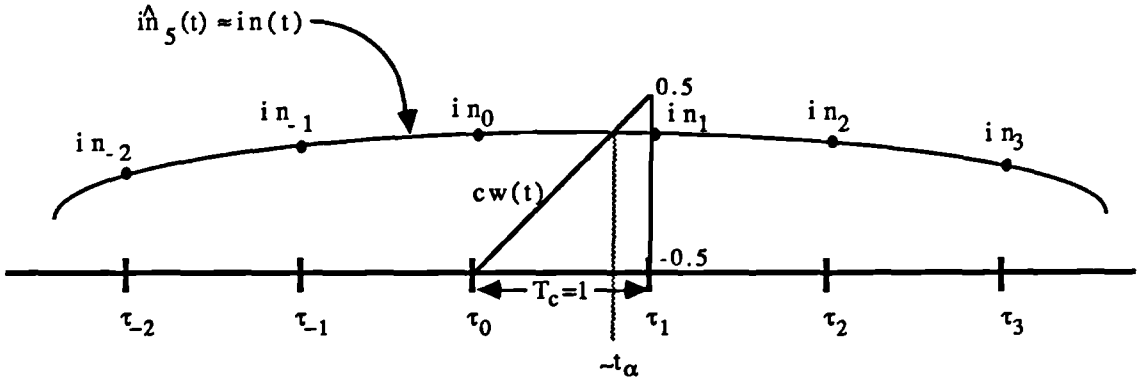


Fig. 5.6: Fifth Order Interpolation Polynomial Approximation to Continuous Time Input, in(t)

polynomial approximations which are shown in Appendix 5C to give to near to 16 bit quality approximation to the cross point time sequence. The Newton form of the interpolation polynomial is used due to the relative computational efficiency with which it may be derived and evaluated. (See Appendix 5B for a discussion of interpolation polynomial approximation in general with a particular emphasis on the Newton polynomial.) These approximations are based on six support points, (in_i,τ_i), $i\in\{-2,-1,\cdots,3\}$ as shown in Fig. 5.6.* The spacing between successive support abscissae is normalized to unity, and the maximum magnitude of the input samples is normalized to 0.5. We denote the fifth order interpolation polynomial approximation to the signal and its derivative as $\hat{in}_5(t)$ and $\hat{in}'_5(t)$, respectively. Hence, we have corresponding fifth order approximations to $f(t)$ and $f'(t)$. They are denoted as $\hat{f}_5(t)$ and $\hat{f}'_5(t)$, respectively, and may be written as (see Appendix 5B):

$$\hat{f}_5(t) = cw(t) - \hat{in}_5(t) \tag{5.16}$$

$$= t - [\, c_{-2} + c_{-1}(t-\tau_{-2}) + c_0(t-\tau_{-2})(t-\tau_{-1}) + \cdots + c_3(t-\tau_{-2})(t-\tau_{-1})\cdots(t-\tau_2)]$$

and

$$\hat{f}'_5(t) = cw'(t) - \hat{in}'_5(t) \tag{5.17}$$

$$= 1 - \left\{ c_{-1} + c_0\left[(t-\tau_{-1})+(t-\tau_{-2})\right] + c_1\left[(t-\tau_{-1})(t-\tau_0)+(t-\tau_{-2})(t-\tau_0)+(t-\tau_{-2})(t-\tau_{-1})\right] \right.$$

$$\left. \cdots + c_3 \sum_{k=-2}^{2} \left[\prod_{\substack{j=-2 \\ j\neq k}}^{2} (t-\tau_j) \right] \right\}$$

$$= 1 - \sum_{i=-1}^{3} c_i \left\{ \sum_{k=-2}^{i-1} \left[\prod_{\substack{j=-2 \\ j\neq k}}^{i-1} (t-\tau_j) \right] \right\}$$

where $\prod_{\substack{j=k \\ j\neq k}}^{k}(t-\tau_j) \equiv 1$. The Newton coefficients, c_i $i\in\{-2,-1,\cdots,3\}$, are obtained from the difference table shown in Table 5.2 where

$$\Delta^j in_i = \Delta^{j-1} in_{i+1} - \Delta^{j-1} in_i, \quad \Delta^0 in_i = in_i \qquad j\in\{1,2,\cdots,5\} \quad i\in\{-2,-1,\cdots,2\} \tag{5.18}$$

* To avoid confusion we have chosen to use τ_i rather than t_i to denote the support abscissae since the later has already been used to denote estimates of the cross point time. Also, we have decided to index the integer, i, from -2 to 3 so that τ_0 and τ_1 will correspond to in_0 and in_1, respectively. This maintains consistency with the indexing for the input samples adopted in Section 5.3.1.

Table 5.2: Difference Table for Signal Approximation Interpolation Polynomial

τ_i	$\Delta^0 in_i$	$\Delta^1 in_i$	$\Delta^2 in_i$	$\Delta^3 in_i$	$\Delta^4 in_i$	$\Delta^5 in_i$
τ_{-2}	$\Delta^0 in_{-2}$					
		$\Delta^1 in_{-2}$				
τ_{-1}	$\Delta^0 in_{-1}$		$\Delta^2 in_{-2}$			
		$\Delta^1 in_{-1}$		$\Delta^3 in_{-2}$		
τ_0	$\Delta^0 in_0$		$\Delta^2 in_{-1}$		$\Delta^4 in_{-2}$	
		$\Delta^1 in_0$		$\Delta^3 in_{-1}$		$\Delta^5 in_{-2}$
τ_1	$\Delta^0 in_1$		$\Delta^2 in_0$		$\Delta^4 in_{-1}$	
		$\Delta^1 in_1$		$\Delta^3 in_0$		
τ_2	$\Delta^0 in_2$		$\Delta^2 in_1$			
		$\Delta^1 in_2$				
τ_3	$\Delta^0 in_3$					

and, with the spacing between successive support abscissae normalized to one:

$$c_i = \frac{\Delta^{i+2} in_{-2}}{(i+2)!} \quad i \in \{-2,-1,\cdots,3\} \tag{5.19}$$

As explained in Appendix 5B the interpolation polynomial and its derivative can be evaluated efficiently with the use of nested multiply methods. Specifically, these may be evaluated as (see Appendix 5B):

$$\hat{in}_5(t) = c'_{-2} \tag{5.20a}$$

and

$$\hat{in}'_5(t) = c'_{-1} + (t-\tau_{-2})\Big[c'_0 + (t-\tau_{-1})\Big[c'_1 + (t-\tau_0)\Big[c'_2 + (t-\tau_1)c'_3\Big]\Big]\Big] \tag{5.20b}$$

and

$$c'_{3-i} \equiv c_{3-i} + (t-\tau_{3-i})c'_{4-i} \quad i \in \{1,2,3,4,5\} \qquad c'_3 \equiv c_3 \tag{5.21}$$

Due to the presence of relatively large approximation errors in $\hat{in}_5(t)$ and $\hat{in}'_5(t)$ over $t \in [\tau_{-2},\tau_{-1}) \cup [\tau_2,\tau_3)$, we compute the three "inner" cross point times over the intervals, $[\tau_{-1},\tau_0)$, $[\tau_0,\tau_1)$, and $[\tau_1,\tau_2)$. (See Appendix 5C for more details.) The interpolation polynomial is therefore updated once every three output samples. Depending on which of these three cross point times we are estimating we set τ_i so as to ensure that the computed

estimate, t_3, is between ±0.5. This is shown in Table 5.3:

Table 5.3: Support Abscissae as a Function of the 3 Cross Point Time Intervals per Polynomial

τ_i	$t_3 \in [\tau_{-1},\tau_0)$	$t_3 \in [\tau_0,\tau_1)$	$t_3 \in [\tau_1,\tau_2)$
τ_{-2}	-1.5	-2.5	-3.5
τ_{-1}	-0.5	-1.5	-2.5
τ_0	0.5	-0.5	-1.5
τ_1	1.5	0.5	-0.5
τ_2	2.5	1.5	0.5
τ_3	3.5	2.5	1.5

5.3.2.2 Computational Complexity

This procedure is computationally intensive. While the exact computation will vary somewhat depending on the precise coding used to implement the procedure, the complexity can be broken down roughly as shown in Table 5.4. The computation required to maintain the difference table and compute the Newton coefficients is common to all three cross points. This is why there are non-integer numbers of instructions per cross point time in the second and third entries in the table.

The overall level of computation is much higher than that of the first procedure. This is due in large part to the need for the additional inversion. However, as we shall see in Chapter Seven, this algorithm yields excellent results.

5.3.3 Cross Point Computation Based on Third Order Approximations

We now describe a cross point algorithm which uses a third order interpolation polynomial. It functions much in the same way as the fifth order procedure of the previous sub-section. For this reason our description will be brief. The algorithm is intended to be a compromise between the fast but crude first order routine and the slow but accurate fifth

order technique.

Table 5.4: Approximate Computational Complexity of the 5th Order Algorithm (Per Cross Point)	
Operation	Computation
compute t_2 (see Section 5.3.1.2)	3 adds, 2 multiplies, 1 inversion
prepare difference table (see Table 5.2)	$\frac{14}{3}$ adds
compute $c_i = [\Delta^{i+2} in_{-2}]\left[\frac{1}{(i+2)!}\right]$ $i \in \{-2,-1,\cdots,3\}$	$\frac{4}{3}$ multiplies
compute $(t_2 - \tau_i)$ $i \in \{-2,-1,\cdots,2\}$	5 adds
compute c'_{3-i} $i \in \{1,2,3,4,5\}$ (see Eq. 5.21)	5 adds 5 multiplies
evaluate $\hat{in}_5(t_2) = c'_{-2}$ (see above)	0
compute $\hat{f}_5(t_2) = [t_2] - [\hat{in}_5(t_2)]$	1 add
evaluate $\hat{in}'_5(t_2)$ (see Eq. 5.20b)	4 adds, 4 multiplies
compute $\hat{f}'_5(t_2) = [1] - [\hat{in}'_5(t_2)]$	1 add
compute $\frac{[1]}{[\hat{f}'_5(t_2)]}$	1 inversion
compute $\left[\hat{f}_5(t_2)\right]\left[\frac{1}{\hat{f}'_5(t_2)}\right]$	1 multiply
compute $t_3 = [t_2] - \left[\frac{\hat{f}_5(t_2)}{\hat{f}'_5(t_2)}\right]$	1 add
Total	$24\frac{2}{3}$ adds, $13\frac{1}{3}$ multiplies, 2 inversions

5.3.3.1 Description of the Algorithm

This procedure is the same as that of Section 5.3.2 except we use a third order interpolation polynomial. In the context of Fig. 5.6, the support points (in_i,τ_i) $i \in \{-1,0,1,2\}$ form the polynomial and, as in the fifth order case, three cross points per polynomial (over $\{[\tau_{-1},\tau_0), [\tau_0,\tau_1), [\tau_1,\tau_2)\}$) are computed.

While the technique will be faster than the fifth order algorithm, the disadvantage, of course, lies in the lower quality of the approximation to the signal and its derivative. An abbreviated error analysis is given in Appendix 5D.

5.3.3.2 Computational Complexity

As mentioned above this procedure represents a computational compromise between the first and fifth order algorithms. The details are given in Table 5.5.

5.4 Alternative Approaches to Cross Point Computation

In the previous section we considered in detail three cross point computation algorithms. Results from computer simulations of the performance of these algorithms will be presented in Chapter Seven. These show that the first, computationally efficient algorithm gives improved (but not distortion free) performance while the second, computationally intensive algorithm gives superior performance. The third algorithm is in between.

It is possible that other techniques can be devised which provide excellent performance at lower levels of computational complexity. In this section we propose (as future work) a few alternative approaches to the cross point computation problem which may be more computationally efficient. It is believed that each of these proposals merit detailed investigation. In particular, three approaches are given. Two are based on the "signal approximation/root finding" method of the previous section and one is based on a nonlinear system identification approach.

Table 5.5: Approximate Computational Complexity of the 3rd Order Algorithm (Per Cross Point)	
Operation	Computation
compute t_2	3 adds, 2 multiplies, 1 inversion
prepare difference table	$\frac{6}{3}$=2 adds
compute $c_i \quad i \in \{-1,0,1,2\}$	$\frac{2}{3}$ multiplies
compute $(t_2-\tau_i) \quad i \in \{-1,0,1\}$	3 adds
compute $c'_{2-i} \quad i \in \{1,2,3\}$	3 adds 3 multiplies
evaluate $\hat{in}_3(t_2) = c'_{-1}$	0
compute $\hat{f}_3(t_2) = [t_2]-[\hat{in}_3(t_2)]$	1 add
evaluate $\hat{in}'_3(t_2)$	2 adds 2 multiplies
compute $\hat{f}'_3(t_2) = [1] - [\hat{in}'_3(t_2)]$	1 add
compute $\frac{[1]}{[\hat{f}'_3(t_2)]}$	1 inversion
compute $\left[\hat{f}_3(t_2)\right]\left[\frac{1}{\hat{f}'_3(t_2)}\right]$	1 multiply
compute $t_3 = [t_2] - \left[\frac{\hat{f}_3(t_2)}{\hat{f}'_3(t_2)}\right]$	1 add
Total	16 adds, $8\frac{2}{3}$ multiplies, 2 inversions

5.4.1 Oversampling

Consider an Nth order interpolation polynomial derived from N+1 equally spaced support abscissae. Intuitively, the accuracy of this polynomial increases over the smallest interval containing the support abscissae as the distance between such abscissae decreases. This closer spacing of support abscissae can be achieved by additional oversampling. As a

specific example, consider using a first order secant algorithm with 16 times oversampling (twice the PWM pulse repetition frequency). This is shown in Fig. 5.7. We first determine between which two of the three samples the cross point lies and then perform the secant iteration using these two samples. Assuming that the errors due to interpolation (i.e., oversampling) are negligible, an estimate of the cross point time based on this technique will be more accurate than the secant iteration as presented in Section 5.3.1. This idea can be extended to higher oversampling factors as well as higher order interpolation polynomials and other root-finding techniques.

Additional computation is of course required as a result of the extra oversampling and any "searching" for the cross point among the additional samples before the actual rootfinding method is applied. The question of whether the increased accuracy is worth the additional computation needs to be evaluated on a case by case basis.

Another approach would be to increase or decrease the pulse repetition frequency, ω_c, so as to try to reduce the overall burden of computation. For example, if we increase ω_c, we reduce the amount of time in which we must compute each cross point time, which is bad. However, as mentioned above, since the support abscissae are closer together the order of the interpolation polynomial required may decrease, which is a computational advantage. Similarly, if we decrease ω_c, we increase the amount of time available in which we compute the cross point time, which is good. But, as the support abscissae are further apart, the order of the interpolation polynomial required to yield approximately 16 bit quality results will increase—of course, a disadvantage. As before the potential for computational savings must be evaluated on a case by case basis. Also, we must be careful not to lower ω_c to the extent that the foldback distortion is no longer negligible. In addition, ω_c should not be increased to the point where serious practical difficulties arise in the construction of the modulator and/or the power switching stage.

5.4.2 Inverse Interpolation

We consider another root finding approach. The formation and evaluation of an *inverse* interpolation polynomial is a well established technique for approximating the solution of a nonlinear equation. Such a polynomial can be used to yield an estimate to the cross point time. Specifically, consider an "extended" PWM comparison waveform, $\overline{cw}(t)\ t \in [\tau_{-2},\tau_3)$, as shown in Fig. 5.8. Also consider a new function, $y=\bar{f}(t)$, the difference between this extended comparison waveform and the analogue input:

$$\bar{f}(t) = \overline{cw}(t) - in(t) \qquad t \in [\tau_{-2},\tau_3) \tag{5.22}$$

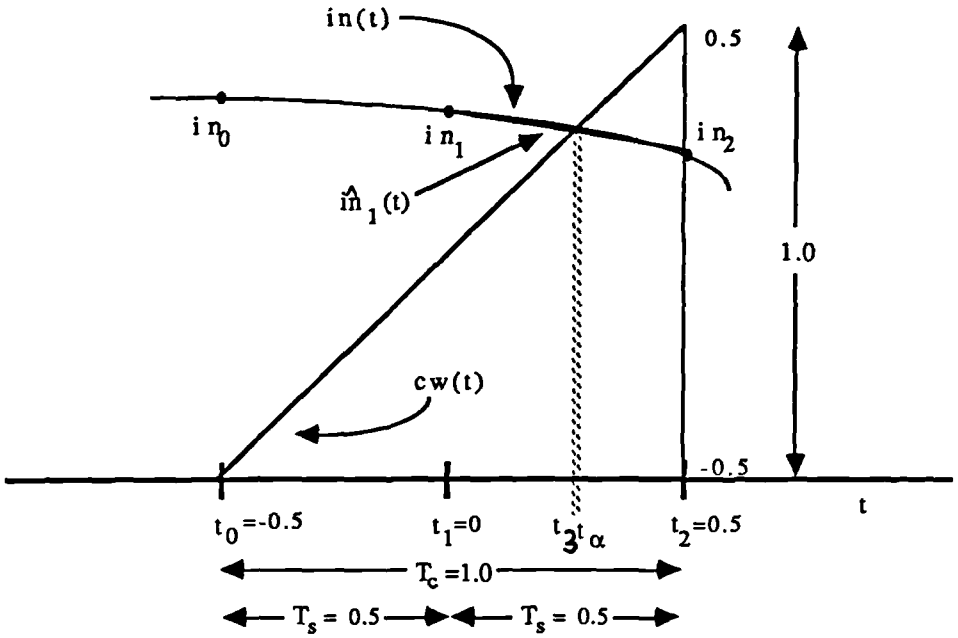


Fig. 5.7: First Order Approximation With Additional Oversampling (16X)

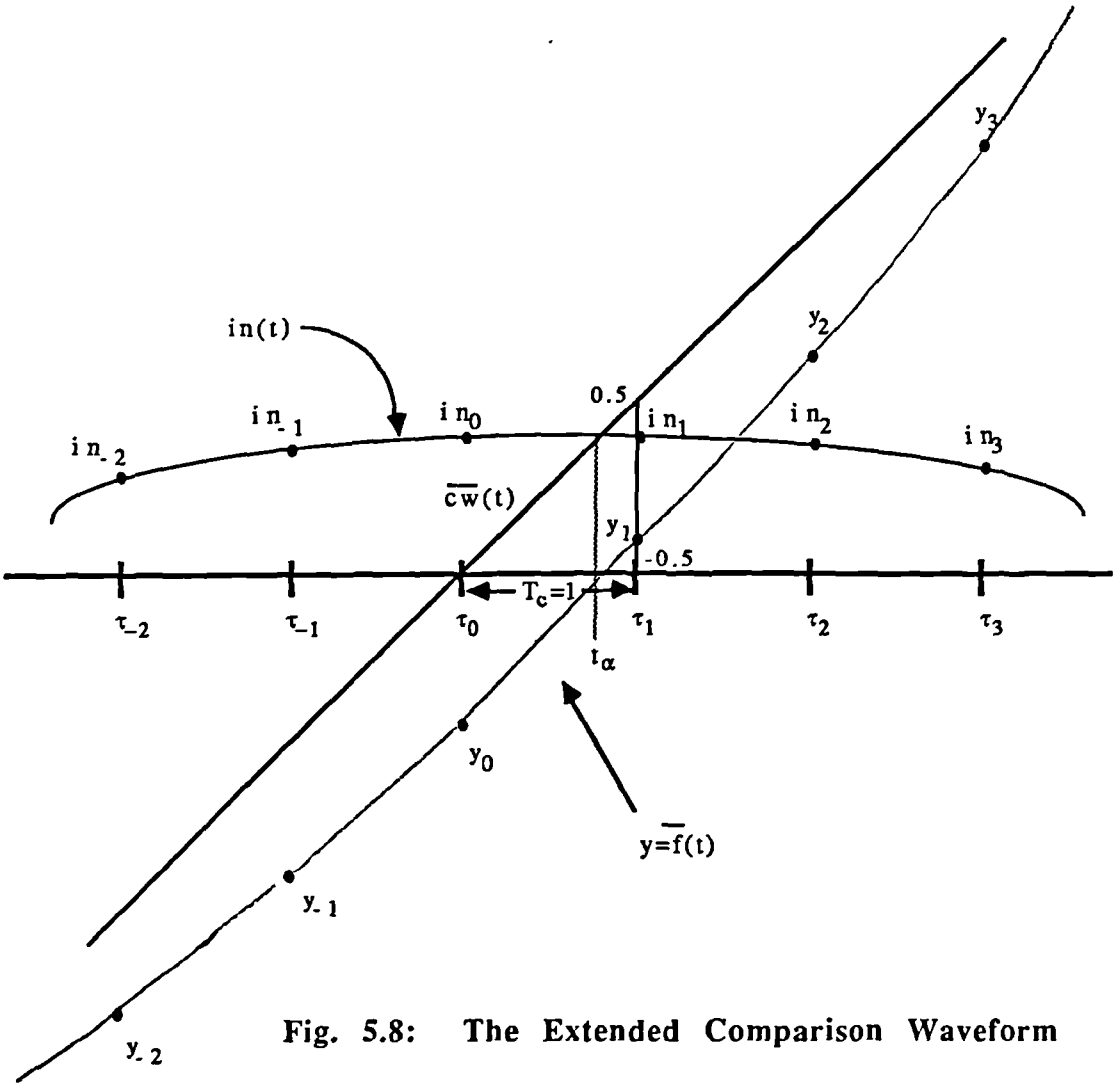


Fig. 5.8: The Extended Comparison Waveform

The inverse function $t = \bar{f}^{-1}(y)$ exists and is meaningful if and only if the "forward" (as opposed to inverse) function, $\bar{f}(t)$, is monotonic on the interval over which we are concerned. (In our application this is always the case since $\bar{f}'(t) > 1 \quad t \in [\tau_{-2}, \tau_3]$.) The Nth order inverse interpolation polynomial, $t = \hat{\bar{f}}_N^{-1}(y)$, satisfies the interpolation conditions by passing through the $N+1$ support points, $(y_i = \bar{f}(\tau_i), t = \tau_i)$, $i \in \{-\frac{1}{2}(N-1), -\frac{1}{2}(N-1)+1, \cdots, \frac{1}{2}(N-1), \frac{1}{2}(N-1)+1\}$ for N odd or $i \in \{-\frac{1}{2}N, -\frac{1}{2}N+1, \cdots, \frac{1}{2}N-1, \frac{1}{2}N\}$ for N even. Such a polynomial can be derived from a divided difference table like the one shown in Table 5B.1 of Appendix 5B for the case of forward interpolation. A typical inverse interpolation polynomial with $N=5$ is shown in Fig. 5.9. An estimate of t_α, the cross point time between τ_0 and τ_1, is given by evaluating the inverse function at $y=0$:

$$t_\alpha \approx \hat{\bar{f}}_5^{-1}(y=0) \tag{5.23}$$

Similarly, adjacent cross point times can be estimated by evaluating the inverse interpolation polynomial at other values of y.

How do we bound the error in the cross point times generated by this technique? There are two sources of error. First is the approximation error associated with the inverse interpolation polynomial. This error is similar to that described by Eq. 5C.7 of Appendix 5C and may be written as:

$$\hat{e}_N(y) = \frac{\Psi_N(y)}{(N+1)!}\left[\hat{\bar{f}}_N^{-1}\right]^{(N+1)}(\eta) \tag{5.24}$$

where η is inside the smallest closed interval containing y and all the support abscissae, y_i (with i as given above), used in the construction of the polynomial. $\left[\hat{\bar{f}}_N^{-1}\right]^{(N+1)}(\cdot)$ denotes the $N+1$th derivative of the inverse function. It is not easy to place a realistic, tight upper bound on $\Psi_N(y)$ (see Eq. 5C.7 of Appendix 5C) since the signal dependent support abscissae are not equally spaced. This is in contrast to the forward interpolation polynomial used in the technique described in Section 5.3. Also, complicated expressions for the $N+1$th derivative of the inverse function present further difficulties in obtaining a realistic bound on the inverse interpolation error. (See [Os73].) The second source of error is that arising from the noise propagation effects in the divided difference table. (See Appendix 5C.) Again, as the support abscissae are not equally spaced bounding this effect is not as easy as in the forward interpolation case.

For both of the above reasons it is felt that trial and error computer simulations may be the most appropriate way to choose the order of the inverse polynomial as well as to evaluate the performance of the system generally.

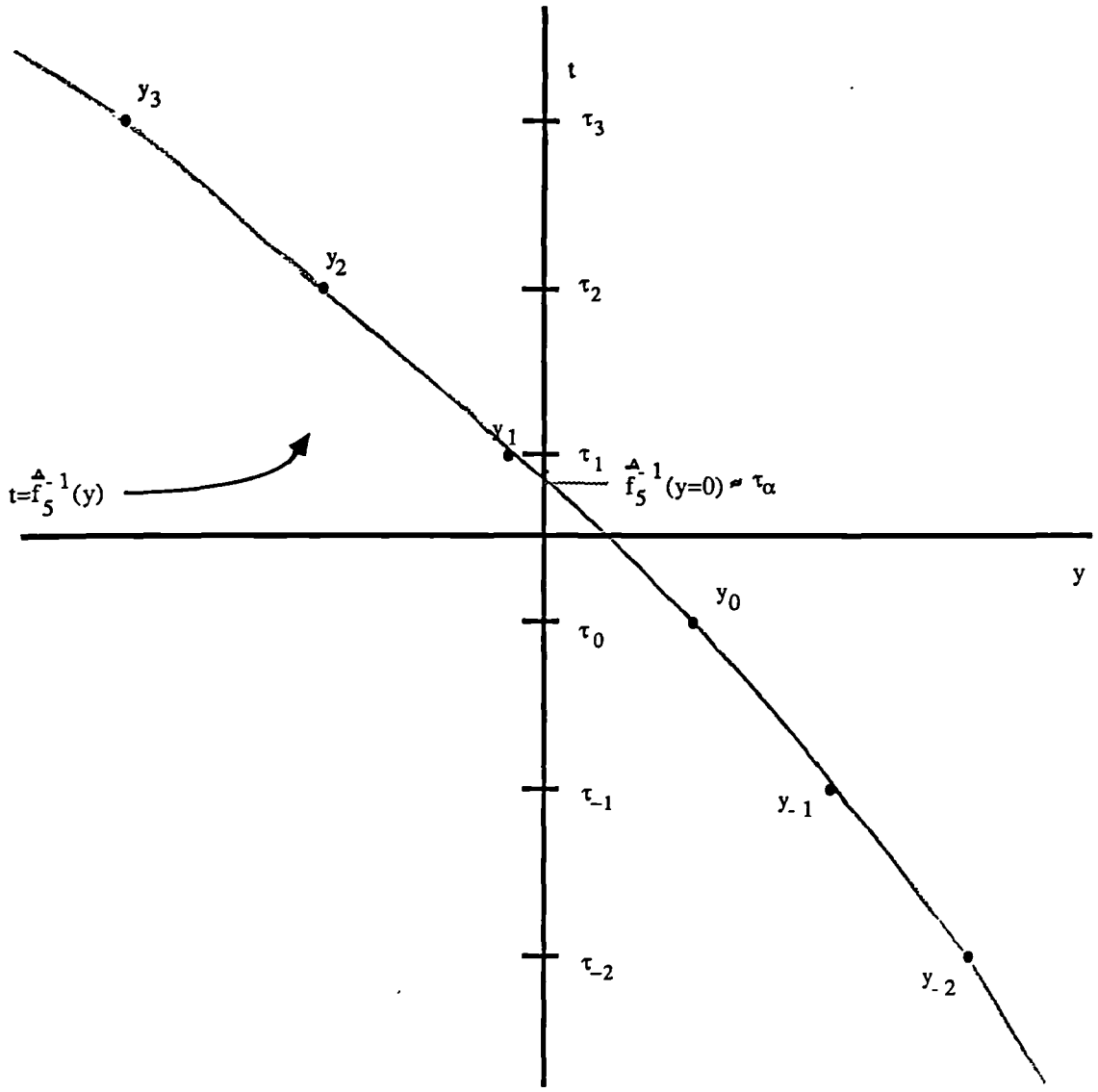


Fig. 5.9: Cross Point Time Estimation via Inverse Interpolation

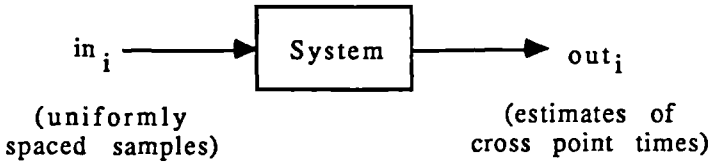


Fig 5.10: Cross Point Deriver as a "Black Box" System

As far as computational complexity is concerned the technique appears attractive as just a single evaluation of a polynomial is required. However, since the support abscissae are not equally spaced we must use a true *divided* difference table rather than a simple (no divisions) difference table as is used in the forward interpolation polynomial of Section 5.3. Thus many computationally intensive *divides* or *inversions* are required to derive the polynomial. Unlike the technique of Section 5.3, the majority of the total computation goes into obtaining the approximation itself rather than any subsequent "root finding" computations. There may be ways of reducing the number of divides by considering alternative methods for deriving the polynomial. (Worthwhile methods may include the so-called Barycentric form of the Lagrange interpolation polynomial as well as the Neville-Aitken method [Sc89].) In addition, it may prove desirable to combine the inverse interpolation approach with that of the previous sub-section. The additional more closely spaced support points resulting from the higher oversampling may permit reductions in the required order of the inverse interpolation polynomial, thus possibly reducing the number of divides.

5.4.3 Nonlinear System Identification

A different approach to the cross point computation problem is that of considering the cross point deriver as a general input-output black box system as shown in Fig. 5.10. We can use system identification methods to characterize the cross point deriver. We know that the cross point deriver is causal, time-invariant, and nonlinear. While the first two properties are readily apparent, the fact that the cross point deriver is nonlinear is verified by showing that scaling the input sequence, in_i, by a constant, K, does not scale the output sequence, out_i, by K. In particular, consider the two adjacent samples, in_0 and in_1, at times τ_0 and τ_1, respectively, of an underlying analogue input, $in(t)$, described by a straight line over the interval of interest:

$$in(t) = mt + b \qquad t \in [\tau_0, \tau_1) \tag{5.25}$$

The cross point time is then the solution of:

$$f(t) = cw(t) - in(t) = t - [mt + b] = 0 \tag{5.26}$$

$$\Rightarrow t = \frac{b}{1 - m}$$

Now let us scale the input by the constant, K. The cross point time is now the solution of:

$$t - K[mt + b] = 0 \tag{5.27}$$

$$\Rightarrow t = \frac{Kb}{1-Km} \neq K\left[\frac{b}{1-m}\right]$$

Thus, scaling the input does not result in a corresponding scaling of the output. In fact, the nonlinearity of the cross point deriver is readily verified by computer simulation. Consider a near full scale, 16 bit, 20kHz sinusoidal input signal with a sampling rate of $f_s = 352.8kHz$. Spectral plots of this input signal and of the corresponding "cross point derived" output signal (i.e., a signal comprised of the cross point times) are given in Figs. 5.11a and 5.11b, respectively. The existence of spectral components in the output at harmonics of input indicate the presence of a nonlinearity. As a full scale, 20kHz tone is the worst case input (i.e., the input resulting in the largest amount of UPWM harmonic distortion) we estimate the effective order of the nonlinearity as seven by counting the number of harmonics above the noise floor in Fig. 5.11b.

So it is clear that when we model the cross point deriver we should use a nonlinear model. One model which has received attention over the past several years is that of the Volterra series expansion where the output of the system is expressed as:

$$y_n = \sum_{i_0=0}^{\infty} h_{i_0} x_{n-i_0} + \sum_{i_0=0}^{\infty}\sum_{i_1=0}^{\infty} h_{i_0 i_1} x_{n-i_0} x_{n-i_1} + \cdots + \sum_{i_0=0}^{\infty}\sum_{i_1=0}^{\infty}\cdots\sum_{i_m=0}^{\infty} h_{i_0 i_1 \cdots i_m} x_{n-i_0} x_{n-i_1} \cdots x_{n-i_m} + \cdots \quad (5.28)$$

with $h_{i_0 i_1 \cdots i_m}$ the mth order "Volterra kernel" [Ru81]. For any practical implementation a truncated form of the Volterra series expansion called a "Volterra filter" is used. The pth order Volterra filter is described by:

$$y_n = \sum_{i_0=0}^{N} h_{i_0} x_{n-i_0} + \sum_{i_0=0}^{N}\sum_{i_1=0}^{N} h_{i_0 i_1} x_{n-i_0} x_{n-i_1} + \cdots + \sum_{i_0=0}^{N}\sum_{i_1=0}^{N}\cdots\sum_{i_p=0}^{N} h_{i_0 i_1 \cdots i_p} x_{n-i_0} x_{n-i_1} \cdots x_{n-i_p} \quad (5.29)$$

The identification problem is then one of estimating the kernels of Eq. 5.29. This is often done using statistical correlation analysis techniques. (See [Ma91] for a large selection of references.)

The computational complexity of the system can increase rapidly with the order of the truncated series. However, techniques have been established to implement Volterra filters more efficiently. (Again see [Ma91] for good references.)

Lastly, as before, it may be possible to combine this approach with those already mentioned. Specifically, refining the output of the first order secant algorithm with a low order Volterra filter placed in cascade may be an option.

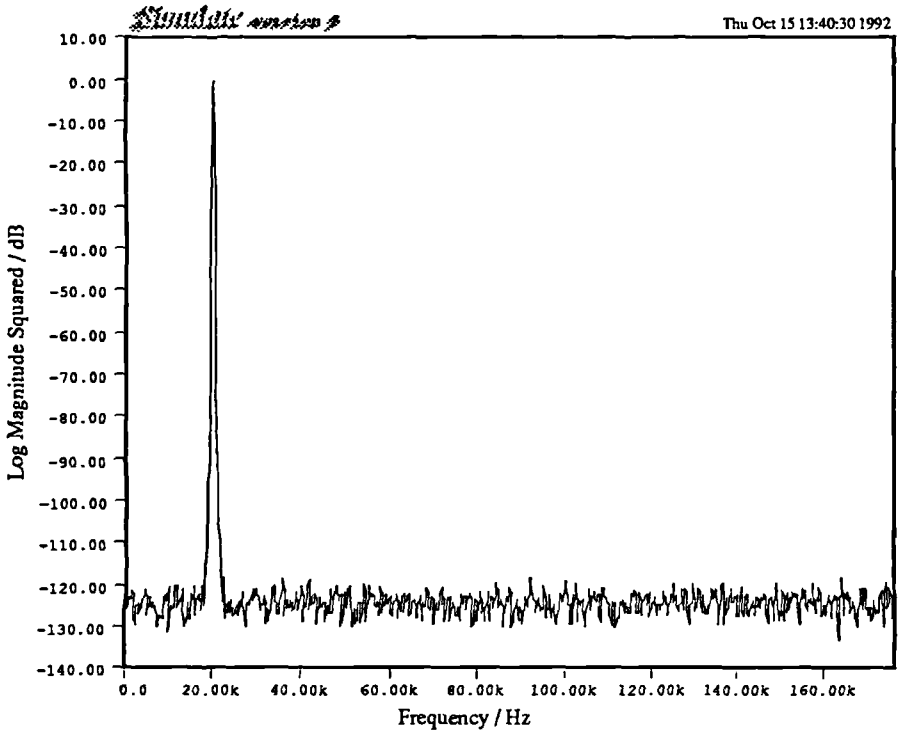


Fig. 5.11a: Tone Input (20kHz, full scale)

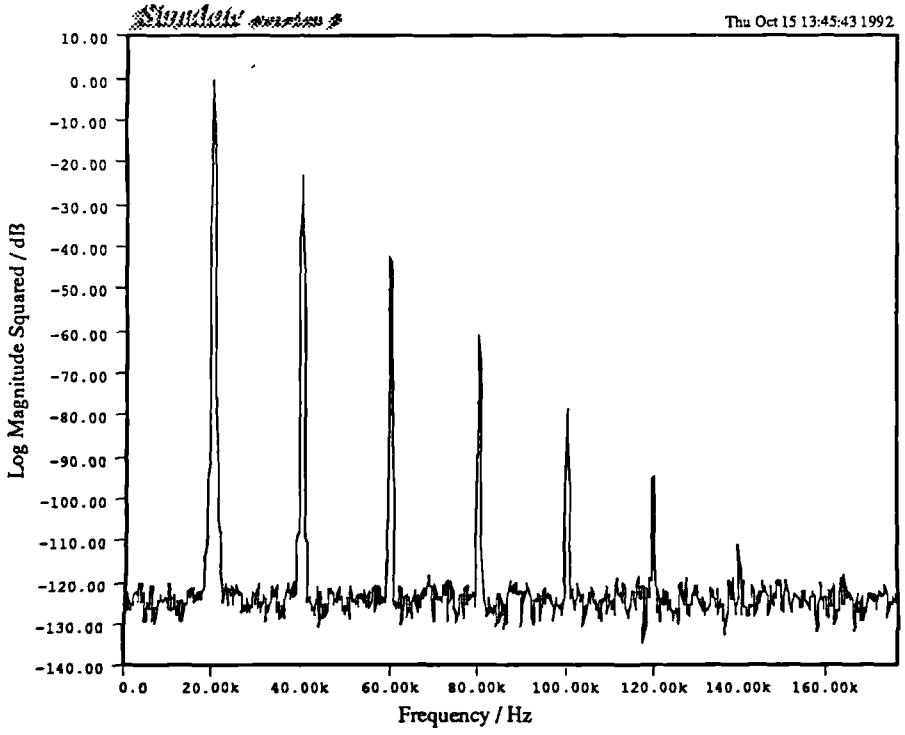


Fig. 5.11b: Cross Point Deriver Output

5.5 Summary

In this chapter we have presented techniques designed to overcome the performance limitations imposed by "standard" UPWM based DACs. The harmonic distortion free performance associated with NPWM was thought to be realizable only in continuous time, analogue systems. However, PNPWM was introduced as a means by which such high levels of performance could be achieved in a discrete time, fully digital system. This is done by estimating the width of the NPWM pulse from the adjacent uniformly spaced digital input samples. Detailed descriptions of three algorithms for a PNPWM based DAC were given. The computational complexity associated with each procedure was analyzed. (Error analyses are given in Appendices 5A, 5C, and 5D.) Several alternative implementations which may be more computationally efficient were also proposed. In Chapter Seven the performance of these procedures will be evaluated by computer simulation.

Appendix 5A: Error Analysis for the First Algorithm

In this appendix we present an error analysis for the procedure described in Section 5.3.1. Ideally the error analyses for all the techniques presented would be based on statistical models for the various sources of error. These would allow us to compute useful quantities such as the average error power or the power spectral density of the error for a given input. However, such an analysis is made difficult by the complicated manner in which the various sources of error are related to each other as well as to the input signal. As a more tractable alternative we derive an upper bound for the error associated with each technique. (The error analyses for the other algorithms are presented in Appendices 5C and 5D.)

The error associated with the first technique can be attributed to two principle sources. The first is what we call the theoretical error, e_{t_s}, and is the amount by which secant approximations differ from the actual root when *infinite* precision arithmetic is used. This is the error inherent to the secant method. The second is an additional error due to the quantization noise on the input signal samples which are, of course, represented with finite precision. We call this additional error e_{q_s}. These errors (which in general may be correlated) combine to give e_{T_s}, the total error associated with this method.

We begin with the theoretical error for the $(j+1)$th iterate. This error (i.e., the difference between the actual solution of the equation and the $(j+1)$th approximation to the solution) can be shown to be given by [At89]:

$$e_{j+1} = t_\alpha - t_{j+1} = -e_j e_{j-1} \frac{f''(\zeta_j)}{2f'(\xi_j)} \qquad \zeta_j \in (t_{j-1}, t_j) \qquad \xi_j \in I_j \qquad j>0 \tag{5A.1}$$

where I_j is the smallest interval containing t_{j-1}, t_j, and the true solution to Eq. 5.3 (i.e., the actual time of the cross point), t_α. Since we know that $t_\alpha \in [t_0, t_1)$ with t_0 and t_1 as shown in Fig. 5.3, we know that $I_1=(t_0,t_1)$. We are interested in placing an upper bound on the magnitude of the error after a single iteration:

$$\left|e_{t_s}\right| = \left|e_1 e_0 \frac{f''(\zeta_1)}{2f'(\xi_1)}\right| \tag{5A.2}$$

To this end consider the product

$$e_1 e_0 = \left[t_\alpha - t_1\right]\left[t_\alpha - t_0\right] = t_\alpha^2 - 0.25 \tag{5A.3}$$

Over $t_\alpha \in [t_0=-0.5, t_1=0.5)$, $|e_1 e_0|$ attains a maximum of 0.25 at $t_\alpha=0$. Also consider a new quantity, M (not to be confused with the modulation depth of Chapter Three):

$$M \equiv \frac{\underset{t\in(t_0,t_1)}{Max}\,|f''(t)|}{\underset{t\in(t_0,t_1)}{Min}\,|2f'(t)|} \geq \left|\frac{f''(\zeta_1)}{2f'(\xi_1)}\right| \qquad \zeta_1\in(t_0,t_1) \quad \xi_1\in I_1 \tag{5A.4}$$

Now for sinusoidal input signals $f(t)$ can be written as:

$$f(t) = cw(t) - in(t) = t - H\sin(\theta t+\phi) \qquad t\in[t_0,t_1) \tag{5A.5}$$

with

$$|H| \leq 0.5, \quad \left|\theta=\frac{2\pi f_b}{f_c}\right| \leq \frac{2\pi 20000}{352800} \approx 0.35619 \tag{5A.6}$$

where f_b (=20kHz) and f_c (=352.8kHz) are the maximum input frequency and the pulse repetition frequency, respectively. Therefore,

$$M = \frac{\underset{t\in(t_0,t_1)}{Max}\,|f''(t)|}{\underset{t\in(t_0,t_1)}{Min}\,|2f'(t)|} = \frac{\underset{t\in(t_0,t_1)}{Max}\,|H\theta^2\sin(\theta t+\phi)|}{\underset{t\in(t_0,t_1)}{Min}\,|2[1-H\theta\cos(\theta t+\phi)]|} \approx \frac{0.5(0.35619)^2}{2[1-0.5(0.35619)]} \approx 3.8591\text{x}10^{-2} \tag{5A.7}$$

From this we can bound the error as:

$$\left|e_{t_2}\right| < Max\left|e_1e_0\frac{f''(\zeta)}{f'(\xi)}\right| < Max\left|e_1e_0\right|M \approx (0.25)(3.8591\text{x}10^{-2}) \tag{5A.8}$$

$$\approx 9.64764\text{x}10^{-3}$$

Next, we consider the error due to quantization of the two input samples, in_0 and in_1, which are used by the secant method. In any real implementation we use *quantized* versions of these samples, which we now denote as $\hat{in}_0$ and $\hat{in}_1$, giving rise to $\hat{t}_2$, a different estimate of the cross point time:

$$\hat{t}_2 = \frac{0.5(\hat{in}_1 + \hat{in}_0)}{1 + \hat{in}_0 - \hat{in}_1} \tag{5A.9}$$

where

$$\hat{in}_0 = in_0+\varepsilon_0 \qquad \hat{in}_1 = in_1+\varepsilon_1 \qquad |\varepsilon_0,\varepsilon_1| \leq \varepsilon_{max} \tag{5A.10}$$

ε_0 and ε_1 are the quantization errors on in_0 and in_1, respectively. ε_{max} is the maximum size of the quantization error on any given sample and is of course determined by the number of bits used to represent the signal at the input to the system. For example, a 16 bit signal with its maximum magnitude normalized to 0.5 implies $\varepsilon_{max} = \frac{0.5}{65536} \approx 7.62939\text{x}10^{-6}$. We have assumed that any additional errors associated with in_0 or in_1 (such as those encountered in the interpolation stage) are negligible. Otherwise these can be added into ε_0 and ε_1.

For the purpose of analysis we can express the cross point time estimate as a product:

$$\hat{t}_2 = \hat{A}\hat{B} \tag{5A.11}$$

where $\hat{A}=0.5(\hat{in}_0+\hat{in}_1)$ and $\hat{B}=\frac{1}{1+\hat{in}_0-\hat{in}_1}$. These two factors themselves can be written as the sum of an "unquantized" term, A (or B), and an error term E_A (or E_B):

$$\hat{A} = A + E_A = 0.5(in_0+in_1) + 0.5(\varepsilon_0+\varepsilon_1) \tag{5A.12}$$

$$\hat{B} = B + E_B = \frac{1}{1+in_0-in_1} + \left[\frac{1}{1+(in_0+\varepsilon_0)-(in_1+\varepsilon_1)} - \frac{1}{1+in_0-in_1}\right] \tag{5A.13}$$

Rewriting Eq. 5A.11 with Eqs. 5A.12 and 5A.13 we obtain:

$$\hat{t}_2 = \hat{A}\hat{B} = \left[A+E_A\right]\left[B+E_B\right] = AB + E_AB + AE_B + E_AE_B \tag{5A.14}$$

e_{t_2} is related to the first term (i.e., $e_{t_2} = t_\alpha - AB$.) Hence this second source of error, e_{q_2}, can be expressed as:

$$e_{q_2} = E_AB + AE_B + E_AE_B \tag{5A.15}$$

A conservative upper bound on $\left|e_{q_2}\right|$ can be obtained by replacing each term in Eq. 5A.15 with its maximum magnitude:

$$\left|e_{q_2}\right| < Max\ |E_A|Max\ |B| + Max\ |A|Max\ |E_B| + Max\ |E_A|Max\ |E_B| \tag{5A.16}$$

It is easy to see that

$$Max\ |A| = 0.5(0.5+0.5) = 0.5 \tag{5A.17}$$

and

$$Max\ |E_A| = 0.5(\varepsilon_{max}+\varepsilon_{max}) = \varepsilon_{max} \approx 7.62939\text{x}10^{-6} \tag{5A.18}$$

Recalling that t_0=–0.5 and t_1=0.5, the maximum magnitude for B is attained when $in(t) = 0.5\sin\left[2\pi\frac{20000}{352800}t\right]$:

$$Max\ |B| = \frac{1}{1 + 0.5\sin\left[2\pi\frac{20000}{352800}(-0.5)\right] - 0.5\sin\left[2\pi\frac{20000}{352800}(0.5)\right]} \tag{5A.19}$$

$$\approx 1.21530$$

Before considering $|E_B|$ we define the following quantities:

$$\delta \equiv -in_0+in_1 \qquad |\delta| \le \sin\left[\frac{2\pi 20000}{352800}(0.5)\right] \approx 0.17715 \tag{5A.20}$$

$$\varepsilon_\delta = -\varepsilon_0+\varepsilon_1 \qquad |\varepsilon_\delta| \le 2\varepsilon_{max} \approx 1.52588\text{x}10^{-5} \tag{5A.21}$$

Hence E_B can be written as:

$$E_B = \frac{1}{1-(\delta+\varepsilon_\delta)} - \frac{1}{1-\delta} \tag{5A.22}$$

$$= \left[1 + (\delta+\varepsilon_\delta) + (\delta+\varepsilon_\delta)^2 + (\delta+\varepsilon_\delta)^3 + \cdots\right] - \left[1 + \delta + \delta^2 + \delta^3 \cdots\right] \tag{5A.23}$$

$$= \varepsilon_\delta + (2\delta\varepsilon_\delta+\varepsilon_\delta^2) + (3\varepsilon_\delta\delta^2+3\varepsilon_\delta^2\delta+\varepsilon_\delta^3) + \cdots$$

Therefore,

$$Max\,|E_B| = Max\left|\varepsilon_\delta + (2\delta\varepsilon_\delta+\varepsilon_\delta^2) + (3\varepsilon_\delta\delta^2+3\varepsilon_\delta^2\delta+\varepsilon_\delta^3) + \cdots\right| \tag{5A.24}$$

We can see that the maximum occurs when δ and ε_δ are at their maximum values. (ε_δ reaches its maximum when $\varepsilon_0 = -\varepsilon_1 = -\varepsilon_{max}$.) As a result we have:

$$Max\,|E_B| \approx \frac{1}{1+\left\{0.5\sin\left[2\pi\frac{20000}{352800}(-0.5)\right]-7.62939\text{x}10^{-6}\right\}-\left\{0.5\sin\left[2\pi\frac{20000}{352800}(0.5)\right]+7.62939\text{x}10^{-6}\right\}} - \frac{1}{1+0.5\sin\left[2\pi\frac{20000}{352800}(-0.5)\right]-0.5\sin\left[2\pi\frac{20000}{352800}(0.5)\right]} \tag{5A.25}$$

$$\approx 2.25370\text{x}10^{-5}$$

Using the numbers from Eqs. 5A.17, 5A.18, 5A.19, and 5A.25 in Eq. 5A.16 yields:

$$\left|e_{q_s}\right| < \sim(7.62939\text{x}10^{-6})(1.21530)+(0.5)(2.25370\text{x}10^{-5}) \tag{5A.26}$$

$$+(7.62939\text{x}10^{-6})(2.25370\text{x}10^{-5}) \approx 2.05407\text{x}10^{-5}$$

Adding this error to the theoretical error we obtain an upper bound for the magnitude of the total error, $\left|e_{T_s}\right|$:

$$\left|e_{T_s}\right| = \left|e_{t_s} + e_{q_s}\right| \le Max\left|e_{t_s}\right| + Max\left|e_{q_s}\right| < 9.66854\text{x}10^{-3} \tag{5A.27}$$

It is interesting to note that the total error is mostly due the theoretical errors associated with the secant method rather than quantization noise effects. Compared to ε_{max}=7.62939x10^{-6}, the quantization error on the input signal, we see that the total error associated with this first algorithm is quite large. However, as is demonstrated in Chapter

Seven application of this simple procedure yields significant improvements over ordinary UPWM.

Appendix 5B: Polynomial Interpolation

In Section 5.3.2.1 we introduced an algorithm where the idea was to use a Newton Raphson iteration on a function, $f(t)$, equal to the difference between the PWM comparison waveform, $cw(t)$, and the underlying analogue signal, $in(t)$ (i.e., $f(t) = cw(t) - in(t)$). This yields a very accurate estimate of the cross point time. (See Appendix 5C.) The Newton Raphson procedure requires values of the function, $f(t)$, and its derivative, $f'(t)$, neither of which are explicitly available in our *discrete time* system. So in order to use the Newton Raphson method we must *approximate* $f(t)$ and $f'(t)$ (or, equivalently, $in(t)$ and $in'(t)$) in some manner.

A computationally efficient approach is that of using an interpolation polynomial and its derivative to approximate $in(t)$ and $in'(t)$, respectively. This polynomial will be a "short time" approximation to the signal and will be updated every few samples. In this appendix we give a brief overview of the aspects of polynomial interpolation which form the basis of our approach.

5B.1 The Basics

We begin by describing how a general Nth order interpolation polynomial approximation to a real valued function, $g(x)$, can be formed. Consider $N+1$ distinct pairs of "support points", (x_i, g_i), $i \in \{0,1,2,\cdots,N\}$ with x_i and g_i the support abscissae and support ordinates, respectively. The Nth order interpolation polynomial:

$$P_N(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N \tag{5B.1}$$

is the Nth order polynomial satisfying the interpolation conditions:

$$g_i = P_N(x_i) \qquad i \in \{0,1,\cdots,N\} \tag{5B.2}$$

This polynomial can be shown to be unique [At89]. Its coefficients, a_i, $i \in \{0,1,\cdots,N\}$, can be determined by solving the system of $N+1$ linear equations in $N+1$ unknowns:

$$\begin{array}{ccccccccc} a_0 & + & a_1 x_0 & + & \cdot & \cdot & \cdot + & a_N x_0^N & = g_0 \\ \cdot & & \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_0 & + & a_1 x_N & + & \cdot & \cdot & \cdot + & a_N x_N^N & = g_N \end{array} \tag{5B.3}$$

The direct solution of this system of equations is computationally intensive. However, other approaches have been developed which are more efficient.

The existence and uniqueness of the interpolation polynomial is often proven by construction [At89, Sch89]. In the course of such proofs it is shown that Eq. 5B.1 can be expressed as a so-called "Lagrange polynomial:"

$$P_N(x) = \sum_{i=0}^{N} g_i l_i(x) \tag{5B.4}$$

where

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{x - x_j}{x_i - x_j} \qquad i \in \{0,1,\cdots,N\} \tag{5B.5}$$

The polynomial of Eq. 5B.4 is numerically equivalent to that of Eq. 5B.1. Another equivalent form of Eq. 5B.1 is the "Newton polynomial" which is written as:

$$P_N(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \cdots + c_N(x-x_0)(x-x_1)\cdots(x-x_N) \tag{5B.6}$$

where c_i, $i \in \{0,1,\cdots,N\}$ are known as the "Newton coefficients." The Newton polynomial can be derived and evaluated with high computational efficiency [Sc89]. This is why it is used in the cross point algorithm presented in Section 5.3.2.1.

The Newton coefficients are obtained from a "divided difference" table such as the one shown in Table 5B.1. $g[x_i, x_{i+1}, \cdots, x_{i+k}]$ is known as the kth divided difference associated with the support abscissae, $x_i, x_{i+1}, \cdots, x_{i+k}$. It is computed recursively as:

$$g[x_i, x_{i+1}, \cdots, x_{i+k}] = \frac{g[x_{i+1}, x_{i+2}, \cdots, x_{i+k}] - g[x_i, x_{i+1}, \cdots, x_{i+k-1}]}{x_k - x_i} \tag{5B.7}$$

with the initial 0th order divided differences given by:

$$g[x_i] = g(x_i) = g_i \tag{5B.8}$$

So, for example,

$$g[x_0, x_1] = \frac{g[x_1] - g[x_0]}{x_1 - x_0} = \frac{g_1 - g_0}{x_1 - x_0} \tag{5B.9}$$

$$g[x_1, x_2] = \frac{g_2 - g_1}{x_2 - x_1}, \qquad g[x_0, x_1, x_2] = \frac{g[x_1, x_2] - g[x_0, x_1]}{x_2 - x_0} \tag{5B.10,5B.11}$$

It can be shown that the Newton coefficients are given by the elements on the uppermost diagonal of entries in the divided difference table [Sc89]:

$$c_k = g[x_0, \cdots, x_k] \quad k \in \{0,1,\cdots,N\} \tag{5B.12}$$

Computation of the Newton coefficients is simplified when the support abscissae are equally spaced. Specifically, let the spacing between successive support abscissae be denoted by the constant, h. In this case, we may construct a *difference table* rather than a

Table 5B.1: Divided Difference Table

x_i	$g[x_i]$	$g[x_i,x_{i+1}]$	$g[x_i,x_{i+1},x_{i+2}]$	. . .
x_0	$g[x_0]$			
		$g[x_0,x_1]$		
x_1	$g[x_1]$		$g[x_0,x_1,x_2]$	
		$g[x_1,x_2]$		. . .
x_2	$g[x_2]$		$g[x_1,x_2,x_3]$	
		$g[x_2,x_3]$		. . .
x_3	$g[x_3]$		$g[x_2,x_3,x_4]$	
		$g[x_3,x_4]$		
x_4	$g[x_4]$		.	
		.	.	
.	.	.	.	
.	.	.		
.	.			

divided difference table as shown in Table 5B.2. $\Delta^k g_i$ is called the kth forward difference of g at x_i and is given by:

$$\Delta^k g_i = \Delta^{k-1} g_{i+1} - \Delta^{k-1} g_i \quad k>0 \tag{5B.13}$$

with

$$\Delta^0 g_i = g(x_i) = g_i \tag{5B.14}$$

It can be shown that the divided differences (and hence the Newton coefficients) are obtained from the forward differences by [At89]:

$$c_k = g[x_0,x_1,\cdots,x_k] = \lambda_k \Delta^k g_0 \qquad k \in \{0,1,\cdots,N\} \tag{5B.15}$$

where

$$\lambda_k \equiv \frac{1}{k!h^k} \qquad k \in \{0,1,\cdots,N\} \tag{5B.16}$$

λ_k can be computed in advance, stored, and multiplied with $\Delta^k g_0$. The forward differences have the big advantage of being computed without the need for computationally intensive *divide* or *invert* instructions.

Table 5B.2: Difference Table

x_i	$\Delta^0 g_i$	$\Delta^1 g_i$	$\Delta^2 g_i$	. . .
x_0	$\Delta^0 g_0$			
		$\Delta^1 g_0$		
$x_1=x_0+h$	$\Delta^0 g_1$		$\Delta^2 g_0$	
		$\Delta^1 g_1$		. . .
$x_2=x_0+2h$	$\Delta^0 g_2$		$\Delta^2 g_1$	
		$\Delta^1 g_2$		. . .
$x_3=x_0+3h$	$\Delta^0 g_3$		$\Delta^2 g_2$	
		$\Delta^1 g_3$		
$x_4=x_0+4h$	$\Delta^0 g_4$		.	
		.	.	
.	.	.	.	
.	.	.		
.	.			

Thus the Newton coefficients can be computed as follows [Sc89]:

for i=0,1,···,N
  $c_i = g_i$

for k=1,2,···,N
  for i=N,N−1,···,k
    $c_i = c_i - c_{i-1}$

for i=0,1,···,N
  $c_i = c_i \lambda_i$

Now recall that the Newton Raphson method requires not only the function whose root we wish to estimate but its derivative as well. An approximation to the latter can be obtained by differentiating the interpolation polynomial:

$$g'(x) \approx P'_N(x) = \frac{d}{dx}\Big[c_0 + c_1(x-x_0) + \cdots + c_N(x-x_0)(x-x_1)\cdots(x-x_{N-1})\Big] \qquad (5B.17)$$

$$= c_1 + c_2\Big[(x-x_1)+(x-x_0)\Big] + c_3\Big[(x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\Big]$$

$$+ \cdots + c_N \sum_{k=0}^{N-1}\left[\prod_{\substack{j=0 \\ j\neq k}}^{N-1}(x-x_j)\right]$$

$$= \sum_{i=1}^{N} c_i \left\{\sum_{k=0}^{i-1}\left[\prod_{\substack{j=0 \\ j\neq k}}^{i-1}(x-x_j)\right]\right\}$$

The numbers of terms in this equation grows quickly as N increases.

5B.2 Efficiency

In this section we discuss how $P_N(x)$ and $P'_N(x)$ can be computed efficiently. Inspection of Eq. 5B.6 indicates that $P_N(x)$ can be evaluated more efficiently by using a Horner-like method of nested multiplies [Sc89]:

$$P_N(x) = c_0+(x-x_0)\Big[c_1+(x-x_1)\Big[c_2+(x-x_2)\Big[\cdots\Big[c_{N-2}+(x-x_{N-2})\Big[c_{N-1}+c_N(x-x_{N-1})\Big]\Big]\cdots\Big]\Big]\Big] \qquad (5B.18)$$

This is a standard approach to efficiently evaluating the Newton polynomial.

The author has not been able to find similar simplifications in the literature for evaluation of the derivative. However, we have derived a procedure for very efficient evaluation of $P'_N(x)$. We begin by defining a set of useful auxiliary quantities, $c'_i \quad i \in \{0,1,\cdots,N\}$, where:

$$c'_{N-i} \equiv c_{N-i} + (x-x_{N-i})c'_{N-i+1} \quad i \in \{1,2,\cdots,N\} \qquad c'_N \equiv c_N \quad (i=0) \qquad (5B.19)$$

(The strange indexing will make the end result easier to use.) So, for example, when i=1 we have:

$$c'_{N-1} = c_{N-1} + (x-x_{N-1})c'_N = c_{N-1} + (x-x_{N-1})c_N \qquad (5B.20)$$

When i=2:

$$c'_{N-2} = c_{N-2} + (x-x_{N-2})c'_{N-1} = c_{N-2} + (x-x_{N-2})\Big[c_{N-1} + (x-x_{N-1})c_N\Big] \qquad (5B.21)$$

Or, when i=N−1:

$$c'_1 = c_1 + (x-x_1)c'_2 = c_1 + (x-x_1)[c_2 + (x-x_2)\,c'_3] \qquad (5B.22)$$

$$= c_1 + (x-x_1)\Big[c_2 + (x-x_2)\Big[\cdots c_{N-2} + (x-x_{N-2})\Big[c_{N-1} + c_N(x-x_{N-1})\Big]\cdots\Big]\Big]$$

We identify these c'_i simply as the terms appearing within the nested multiplies of Eq. 5B.18, the expression for the efficient evaluation of $P_N(x)$. In fact, it can be seen that $c'_0 = P_N(x)$.

It is easiest to see how we can use c'_i to increase the efficiency with which $P'_N(x)$ can be evaluated for a specific example. Consider the case of N=5 (which is the order of the polynomial used in the algorithm presented in Section 5.3.2). We begin be expanding out the auxiliary quantities defined in Eq. 5B.19.

$$c'_5 = c_5 \tag{5B.23a}$$

$$c'_4 = c_4 + (x-x_4)c'_5 = c_4 + (x-x_4)c_5 \tag{5B.23b}$$

$$c'_3 = c_3 + (x-x_3)c'_4 = c_3 + (x-x_3)\Big[c_4 + (x-x_4)c_5\Big] \tag{5B.23c}$$

$$c'_2 = c_2 + (x-x_2)c'_3 = c_2 + (x-x_2)\Big[c_3 + (x-x_3)\Big[c_4 + (x-x_4)c_5\Big]\Big] \tag{5B.23d}$$

$$c'_1 = c_1 + (x-x_1)c'_2 = c_1 + (x-x_1)\Big[c_2 + (x-x_2)\Big[c_3 + (x-x_3)\Big[c_4 + (x-x_4)c_5\Big]\Big]\Big] \tag{5B.23e}$$

$$c'_0 = c_0 + (x-x_0)c'_1 \tag{5B.23f}$$

$$= c_0 + (x-x_0)\Big[c_1 + (x-x_1)\Big[c_2 + (x-x_2)\Big[c_3 + (x-x_3)\Big[c_4 + (x-x_4)c_5\Big]\Big]\Big]\Big]$$

We observe that these c'_i are just the intermediate quantities automatically generated by the nested multiply structure used to efficiently evaluate $P_5(x)$. (See Eq. 5B.18.) We also see from Eq. 5B.23f and 5B.18 that, as expected, $c'_0 = P_5(x)$.

For the derivative, from 5B.17 we have:

$$P'_5(x) = \sum_{i=1}^{5} c_i \left\{ \sum_{k=0}^{i-1} \left[\prod_{\substack{j=0 \\ j \neq k}}^{i-1} (x - x_j) \right] \right\} \tag{5B.24}$$

$$= c_1 + c_2\left[(x-x_1)+(x-x_0)\right] + c_3\left[(x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\right]$$

$$+ c_4\left[(x-x_1)(x-x_2)(x-x_3)+(x-x_0)(x-x_2)(x-x_3)+(x-x_0)(x-x_1)(x-x_3)+(x-x_0)(x-x_1)(x-x_2)\right]$$

$$+ c_5\Big[(x-x_1)(x-x_2)(x-x_3)(x-x_4)+(x-x_0)(x-x_2)(x-x_3)(x-x_4)+(x-x_0)(x-x_1)(x-x_3)(x-x_4)$$

$$+(x-x_0)(x-x_1)(x-x_2)(x-x_4)+(x-x_0)(x-x_1)(x-x_2)(x-x_3)\Big]$$

This can be expressed as:

$$P'_5(x) = c_1 + c_2(x-x_1) + c_3(x-x_1)(x-x_2) + c_4(x-x_1)(x-x_2)(x-x_3) \tag{5B.25}$$
$$+c_5(x-x_1)(x-x_2)(x-x_3)(x-x_4)$$
$$+ c_2(x-x_0) + c_3(x-x_0)(x-x_2) + c_4(x-x_0)(x-x_2)(x-x_3) + c_5(x-x_0)(x-x_2)(x-x_3)(x-x_4)$$
$$+ c_3(x-x_0)(x-x_1) + c_4(x-x_0)(x-x_1)(x-x_3) + c_5(x-x_0)(x-x_1)(x-x_3)(x-x_4)$$
$$+ c_4(x-x_0)(x-x_1)(x-x_2) + c_5(x-x_0)(x-x_1)(x-x_2)(x-x_4)$$
$$+ c_5(x-x_0)(x-x_1)(x-x_2)(x-x_3)$$

Moreover, rewriting each line of the above Eq. 5B.25 using nested multiplies where possible we obtain:

$$P'_5(x) = c_1 + (x-x_1)\Big[c_2 + (x-x_2)\Big[c_3 + (x-x_3)\Big[c_4 + (x-x_4)c_5\Big]\Big]\Big] \tag{5B.26}$$
$$+ (x-x_0)\Big[c_2 + (x-x_2)\Big[c_3 + (x-x_3)\Big[c_4 + (x-x_4)c_5\Big]\Big]\Big]$$
$$+ (x-x_0)(x-x_1)\Big[c_3 + (x-x_3)\Big[c_4 + (x-x_4)c_5\Big]\Big]$$
$$+ (x-x_0)(x-x_1)(x-x_2)\Big[c_4 + (x-x_4)c_5\Big]$$
$$+ (x-x_0)(x-x_1)(x-x_2)(x-x_3)c_5$$

We recognize the nested terms as the c'_i defined in Eq. 5B.19. Hence,

$$P'_5(x) = c'_1 + c'_2(x-x_0) + c'_3(x-x_0)(x-x_1) + c'_4(x-x_0)(x-x_1)(x-x_2) \tag{5B.27}$$
$$+ c'_5(x-x_0)(x-x_1)(x-x_2)(x-x_3)$$

This itself can of course be written in a nested form:

$$P'_5(x) = c'_1 + (x-x_0)\Big[c'_2 + (x-x_1)\Big[c'_3 + (x-x_2)\Big[c'_4 + (x-x_3)c'_5\Big]\Big]\Big] \tag{5B.28}$$

which is similar in structure to the expression for the efficient evaluation of the function (Eq. 5B.18).

Thus the Newton polynomial and its derivative can be evaluated efficiently by using the procedures:

Compute Newton coefficients $c_i \quad i \in \{0,1,\cdots,N\}$
(see procedure outlined after Eq. 5B.16)

Compute and store $c'_i \quad i \in \{0,1,\cdots,N\}$

$c'_N = c_N$
for i=1,2,$\cdots$,N
$\quad c'_{N-i} = c_{N-i} + (x-x_{N-i})c'_{N-i+1}$

Evaluate $P_N(x)$ (or P) (in effect already done above)

$P = c'_0$

Evaluate $P'_N(x)$ (or P')

$P' = c'_N$
for i=N-1,N-1,$\cdots$1
$\quad P' = c'_i + (x-x_{i-1})P'$

The above algorithm is very efficient. Storing c'_i is not something which is normally done when evaluating just the polynomial itself. However, as we have seen, the derivative can be computed with a minimum amount of computation if these intermediate quantities are

retained. In fact, in spite of the formidable size of Eq. 5B.24, evaluation of the derivative requires even less computation than the polynomial itself! An assessment of the overall computational complexity for the fifth order case is given in Table 5B.3.

Table 5B.3: Fifth Order Polynomial Computational Complexity

Operation	Computation
prepare difference table (see Table 5B.2)	15 adds
compute $(x-x_i)$ $i \in \{0,1,\cdots,4\}$	5 adds
compute c_i $i \in \{0,1,\cdots,5\}$ (see Eq. 5B.15)	4 multiplies
compute c'_i $i \in \{0,1,\cdots 5\}$ (Eq. 5B.15)	5 adds 5 multiplies
compute $P_5(x) = c'_0$	0 see previous row
compute $P'_5(x)$ (see Eq. 5B.28)	4 adds, 4 multiplies

Appendix 5C: Error Analysis for the Fifth Order Algorithm

In this appendix we present an error analysis for the cross point computation algorithm presented in Section 5.3.2.1. Recall that the technique consists of applying a Newton-Raphson iteration on the estimate generated by the crude first order algorithm of Section 5.3.1.1 to yield an accurate approximation of the cross point time. There are three basic sources of error in this algorithm. First are the "theoretical errors" inherent to the Newton-Raphson root-finding technique. Second are the "approximation errors" associated with the use of finite order polynomials to approximate the input signal and its derivative. Last are the "quantization noise propagation errors" which are due to the 16 bit quantization noise on the input signal samples used to form the polynomial approximation to the signal and its derivative.

We begin with $e_{t_{nr}}$, the theoretical error generated by a single Newton Raphson iteration. In general the error associated with the j+1th Newton Raphson iteration (i.e., the difference between the j+1th iterate, t_{j+1}, and the exact solution, t_α) can be shown to be given by [At89]:

$$e_{j+1} = t_\alpha - t_{j+1} = -e_j^2 \frac{f''(\zeta_j)}{2f'(t_j)} \qquad \zeta_j \in I_j \tag{5C.1}$$

with ζ_j a function of t_j and I_j the smallest interval including t_j and t_α. Recall from Appendix 5A that the secant algorithm generated an estimate, $\hat{t}_2$, with total error of magnitude:

$$\left|e_{T_s}\right| = \left|t_\alpha - \hat{t}_2\right| \le 9.66854\text{x}10^{-3} \tag{5C.2}$$

Hence the magnitude of the theoretical error associated with the Newton-Raphson iteration is given by:

$$\left|e_{t_{nr}}\right| = e_{T_s}^2 \frac{|f''(\zeta_2)|}{|2f'(\hat{t}_2)|} \tag{5C.3}$$

An upper bound may be written as:

$$\left|e_{t_{nr}}\right| \le Max\ (e_{T_s}^2)(M) \approx (9.66854\text{x}10^{-3})^2(3.85905\text{x}10^{-2}) \approx 3.60747\text{x}10^{-6} \tag{5C.4}$$

with M as defined in Eq. 5A.4 of Appendix 5A. This is slightly less than $\varepsilon_{max} \approx 7.62939\text{x}10^{-6}$, the maximum magnitude of the quantization error associated with a 16 bit signal normalized to 0.5.

Next we consider the approximation errors which result from representing $f(t)$ and $f'(t)$ with Nth order interpolation polynomial approximations. Recall that:

$$f(t) = cw(t) - in(t) = t - in(t) \tag{5C.5}$$

$$f'(t) = cw'(t) - in'(t) = 1 - in'(t) \tag{5C.6}$$

So any polynomial approximation to $f(t)$ or $f'(t)$ can equivalently be thought of as approximations to $in(t)$ or $in'(t)$, respectively. Also recall that in general $\hat{in}_N(t)$, an Nth order interpolation polynomial approximation to $in(t)$, is derived from the $N+1$ support points, (in_i,τ_i), with $i \in [-½(N-1),-½(N-1)+1,\cdots,½(N-1),½(N-1)+1]$ for N odd or $i \in [-½N,-½N+1,\cdots,½N-1,½N]$ for N even. The approximation to the derivative of the signal is obtained by simply differentiating the approximation to the signal itself. The quality of these approximations can be assessed by bounding the maximum error in the polynomial and its derivative. From standard results in interpolation theory [At89]:

$$\hat{e}_N(t) \equiv f(t) - \hat{f}_N(t) = \frac{\Psi_N(t)}{(N+1)!} f^{(N+1)}(\eta) \qquad t,\eta \in H \tag{5C.7}$$

$$\hat{e}'_N(t) \equiv f'(t) - \hat{f}'_N(t) = \frac{\Psi'_N(t)}{(N+1)!} f^{(N+1)}(\gamma_1) + \frac{\Psi_N(t)}{(N+2)!} f^{(N+2)}(\gamma_2) \qquad t,\gamma_1,\gamma_2 \in H \tag{5C.8}$$

where for N odd $\Psi_N(t) = \prod_{-½(N-1)}^{½(N-1)+1} (t-\tau_i)$ and $H=[\tau_{-½(N-1)},\tau_{½(N-1)+1}]$. For N even, $\Psi_N(t) = \prod_{-½N}^{½N} (t-\tau_i)$ and $H=[\tau_{-½N},\tau_{½N}]$.

Recall that we are using fifth order approximations. Plots of $\Psi_5(t)$ and $\Psi'_5(t)$ for τ_i equally spaced with $\tau_i = -2.5+(i+2)$, $i \in \{-2,-1,0,1,2,3\}$ are shown in Fig. 5C.1. From the figure (and the computer results used to generate the figure) it is seen that:

$$\max_{t \in [\tau_{-1},\tau_2]} \left|\Psi_5(t)\right| \le \sim 5.049 \qquad \max_{t \in [\tau_{-1},\tau_2]} \left|\Psi'_5(t)\right| \le 24.000 \tag{5C.9}$$

Hence,

$$|\hat{e}_5(t)| \le \max_{t \in [\tau_{-1},\tau_2]} \left|\frac{\Psi_5(t)}{6!} f^{(6)}(t)\right| \le \sim 7.160\text{x}10^{-6} \tag{5C.10}$$

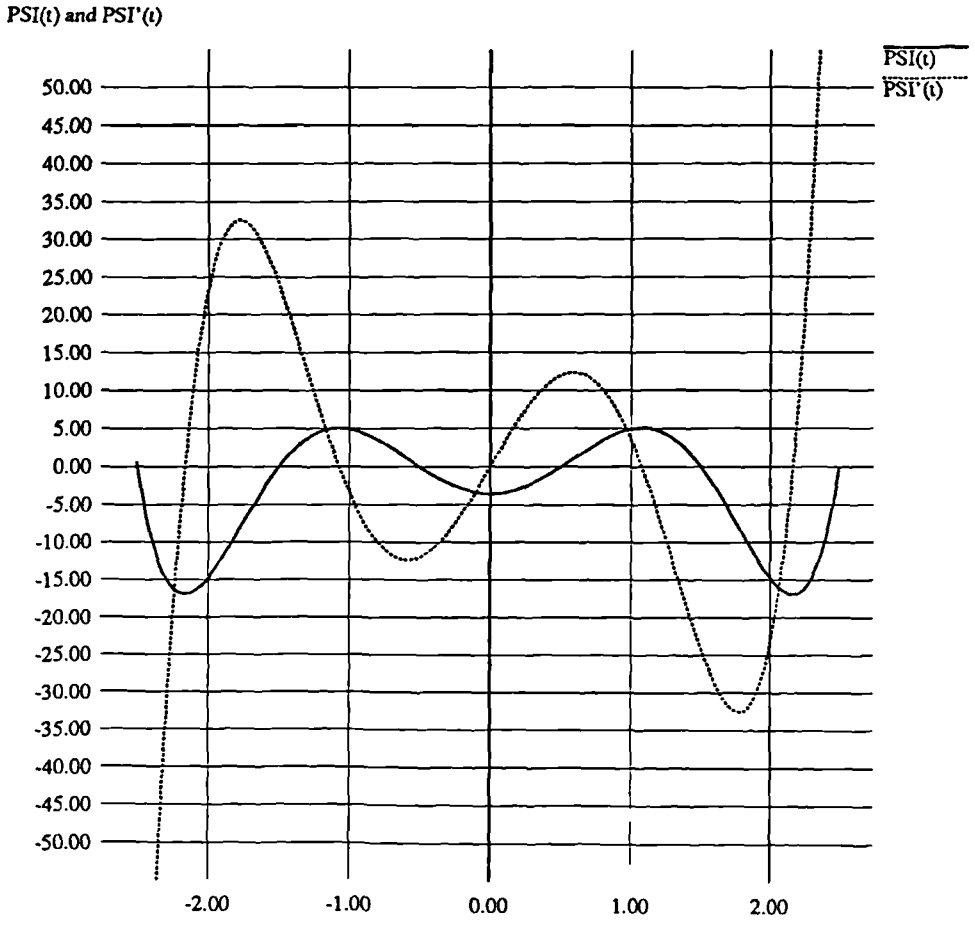
$$|\hat{e}'_5(t)| \le \max_{t \in [\tau_{-1},\tau_2]} \left|\frac{\Psi'_5(t)}{6!} f^{(6)}(t)\right| + \max_{t \in [\tau_{-1},\tau_2]} \left|\frac{\Psi_5(t)}{7!} f^{(7)}(t)\right| \tag{5C.11}$$

$$< \sim 3.40358\text{x}10^{-5} + \sim 3.64418\text{x}10^{-7} \approx 3.44002\text{x}10^{-5}$$

Interestingly, we see that the second term in Eq. 5C.11 is much smaller than the first.

Note the restricted interval, $[\tau_{-1},\tau_2) \subset H$. As mentioned in Section 5.3.2.1 we limit computation to the middle three cross points of each polynomial as large oscillations in

Fig. 5C.1: PSI(t) and PSI'(t) (N=5)



$\Psi_5(t)$ and $\Psi'_5(t)$ over $[\tau_{-2},\tau_{-1})\cup[\tau_2,\tau_3)$ result in large increases in the approximation error for both the signal and its derivative.

We next turn to the propagation of the quantization noise on the input samples used to form the polynomial. In any real (i.e., finite word length) implementation rather than using in_i the sample values shown earlier in the difference table, we must use *quantized* versions which we write as a sum of the unquantized sample plus noise:

$$\hat{in}_i \equiv in_i + \varepsilon_i, \quad |\varepsilon_i| \le \varepsilon_{max} \tag{5C.12}$$

The quantization errors on the input signal propagate through the table and result in errors on the coefficients of the Newton polynomial. This is shown in Table 5C.1:

Table 5C.1: Divided Difference Table with Quantization Effects

τ_i	$\hat{in}[\tau_i]$	$\hat{in}[\tau_i,\tau_{i+1}]$	$\hat{in}[\tau_i,\tau_{i+1},\tau_{i+2}]$ · · ·
τ_{-2}	$\hat{in}[\tau_{-2}]=(in_{-2}+\varepsilon_{-2}$		
	$=in_{-2}+\varepsilon_{c_{-2}})$	$\hat{in}[\tau_{-2},\tau_{-1}]=(in[\tau_{-2},\tau_{-1}]+\varepsilon_{-1}-\varepsilon_{-2}$	
τ_{-1}	$\hat{in}[\tau_{-1}]=(in_{-1}+\varepsilon_{-1})$	$=in[\tau_{-2},\tau_{-1}]+\varepsilon_{c_{-1}})$	$\hat{in}[\tau_{-2},\tau_{-1},\tau_0]=(in[\tau_{-2},\tau_{-1},\tau_0]+\frac{1}{2}\{\varepsilon_0-2\varepsilon_{-1}+\varepsilon_{-2}\}$
		$\hat{in}[\tau_{-1},\tau_0]=(in[\tau_{-1},\tau_0]+\varepsilon_0-\varepsilon_{-1})$	$=in[\tau_{-2},\tau_{-1},\tau_0]+\varepsilon_{c_0})$ · · ·
τ_0	$\hat{in}[\tau_0]=(in_0+\varepsilon_0)$		$\hat{in}[\tau_{-1},\tau_0,\tau_1]=(in[\tau_{-1},\tau_0,\tau_1]+\frac{1}{2}\{\varepsilon_1-2\varepsilon_0+\varepsilon_{-1}\})$
		$\hat{in}[\tau_0,\tau_1]=(in[\tau_0,\tau_1]+\varepsilon_1-\varepsilon_0)$	· · ·
τ_1	$\hat{in}[\tau_1]=(in_1+\varepsilon_1)$		$\hat{in}[\tau_0,\tau_1,\tau_2]=(in[\tau_0,\tau_1,\tau_2]+\frac{1}{2}\{\varepsilon_2-2\varepsilon_1+\varepsilon_0\})$
		$\hat{in}[\tau_1,\tau_2]=(in[\tau_1,\tau_2]+\varepsilon_2-\varepsilon_1)$	
τ_2	$\hat{in}[\tau_2]=(in_2+\varepsilon_2)$		
		$\hat{in}[\tau_2,\tau_3]=(in[\tau_2,\tau_3]+\varepsilon_3-\varepsilon_2)$	
τ_3	$\hat{in}[\tau_3]=(in_3+\varepsilon_3)$		

(We show the divided difference table rather than the associated difference table to more explicitly show the errors on the Newton coefficients themselves.) We can see from the table that the computed coefficients can be written as a sum of the true (i.e., infinite precision) coefficients plus error terms:

$$\hat{in}[\tau_{-2},\cdots,\tau_i] = \hat{c}_i \equiv c_i + \varepsilon_{c_i} \qquad i \in \{-2,-1,\cdots,3\} \tag{5C.13}$$

where each ε_{c_i} is a function ε_j $j \in \{-2,-1,\cdots,i\}$ in the table. Therefore, we can express the actual interpolation polynomial we use as a sum of the true analogue input plus an approximation error and a noise polynomial with a similar sum for the derivative of the interpolation polynomial:

$$\hat{\hat{in}}_5(t) \equiv in(t) + \hat{e}_5(t) + \hat{\varepsilon}_c(t) \tag{5C.14}$$

$$\hat{\hat{in}}'_5(t) \equiv in'(t) + \hat{e}'_5(t) + \hat{\varepsilon}'_c(t) \tag{5C.15}$$

with

$$\hat{\varepsilon}_c(t) = \varepsilon_{c_{-2}} + \varepsilon_{c_{-1}}(t-\tau_{-2}) + \cdots + \varepsilon_{c_3}[(t-\tau_{-2})\cdots(t-\tau_2)] \tag{5C.16}$$

It has been shown by computer simulation that:

$$\underset{\substack{t\in[\tau_{-1},\tau_2) \\ |\varepsilon_i|\le\varepsilon_{max},\ i\in\{-2,-1,\cdots,3\}}}{Max} \left|\hat{\varepsilon}_c(t)\right| \le -1.24040\text{x}10^{-5} \tag{5C.17}$$

$$\underset{\substack{t\in[\tau_{-1},\tau_2)\\ |\varepsilon_i|\le\varepsilon_{max},\ i\in\{-2,-1,\cdots,3\}}}{Max}\left|\hat{\varepsilon}'_c(t)\right| \le -3.56038\text{x}10^{-5} \tag{5C.18}$$

We can now express $\hat{t}_3$, the actual computed cross point time generated by the algorithm as:

$$\hat{t}_3 = \hat{t}_2 - \frac{\hat{t}_2 - \hat{in}_5(\hat{t}_2)}{1-\hat{in}'_5(\hat{t}_2)} \tag{5C.19}$$

To place an upper bound on the overall error we use the approach taken in the error analysis for the first technique by restating the above equation as:

$$\hat{t}_3 = \hat{t}_2 - \hat{C}\hat{D} \tag{5C.20}$$

where $\hat{C}=\hat{t}_2-\hat{in}_5(\hat{t}_2)$ and $\hat{D}=\dfrac{1}{1-\hat{in}'_5(\hat{t}_2)}$. As in Appendix 5A these two terms can be written as a sum of an "ideal" term (i.e., without approximation and quantization noise propagation errors) and an error term. Specifically, we have:

$$\hat{C} = C + E_C = [\hat{t}_2 - in(\hat{t}_2)] + [-\hat{e}_5(\hat{t}_2) - \hat{\varepsilon}_c(\hat{t}_2)] \tag{5C.21}$$

$$\hat{D} = D + E_D = \left[\frac{1}{1-in'(\hat{t}_2)}\right] + \left[\frac{1}{1-[in'(\hat{t}_2)+\hat{e}'_5(\hat{t}_2)+\hat{\varepsilon}'_c(\hat{t}_2)]} - \frac{1}{1-in'(\hat{t}_2)}\right] \tag{5C.22}$$

Therefore,

$$\hat{t}_3 = \hat{t}_2 - \Big[C+E_C\Big]\Big[D+E_D\Big] = \hat{t}_2 - (CD + E_C D + E_D C + E_C E_D) \tag{5C.23}$$

Hence, the effect of the sum of the approximation error $e_{a_{nr}}$ and the quantization noise propagation error $e_{q_{nr}}$ on the final result can be written as:

$$e_{a_{nr}} + e_{q_{nr}} = -(E_C D + E_D C + E_C E_D) \tag{5C.24}$$

As in Appendix 5A an upper bound for the magnitude of this error can be found by replacing each term in the above equation with its maximum magnitude:

$$\left|e_{a_{nr}}+e_{q_{nr}}\right| < Max\,|E_C|\,Max\,|D| + Max\,|E_D|\,Max\,|C| + Max\,|E_C|\,Max\,|E_D| \tag{5C.25}$$

It is easily seen from the end of Appendix 5A that:

$$Max\,|C| = Max\,|\hat{t}_2 - in(\hat{t}_2)| < e_{T_s} \approx 9.66854\text{x}10^{-3} \tag{5C.26}$$

From Eqs. 5C.10 and 5C.17 we have:

$$Max\,|E_C| = Max\,|-\hat{e}_5(\hat{t}_2) - \hat{\varepsilon}_c(\hat{t}_2)| < Max\,|\hat{e}_5(\hat{t}_2)| + Max\,|\hat{\varepsilon}_c(\hat{t}_2)| \tag{5C.27}$$

$$\approx 7.16029\text{x}10^{-6} + 1.24040\text{x}10^{-5} \approx 1.95643\text{x}10^{-5}$$

Also,

$$Max\,|D| = Max\left|\frac{1}{1-in'(\hat{t}_2)}\right| \approx \left[\frac{1}{1-(0.5)(0.35619)}\right] \approx 1.21669 \qquad (5C.28)$$

From arguments similar to those given for $Max\,|E_B|$ in Appendix 5A it is seen that:

$$Max\,|E_D| = Max\left|\frac{1}{1-\{in'(\hat{t}_2)+\hat{e}_5(\hat{t}_2)+\hat{e}'_c(\hat{t}_2)\}} - \frac{1}{1-in'(\hat{t}_2)}\right| \qquad (5C.29)$$

$$< \sim \frac{1}{1-\{(0.5)(0.35619)+3.44002\text{x}10^{-5}+3.56038\text{x}10^{-5}\}} - \frac{1}{1-(0.5)(0.35619)}$$

$$\approx 1.03637\text{x}10^{-4}$$

Using Eqs. 5C.26 to 5C.29 in Eq. 5C.25 yields:

$$\left|e_{a_{nr}}+e_{q_{nr}}\right| < (1.95643\text{x}10^{-5})(1.21669) + (1.03637\text{x}10^{-4})(9.66854\text{x}10^{-3}) \qquad (5C.30)$$

$$+ (1.95643\text{x}10^{-5})(1.03637\text{x}10^{-4}) \approx 2.48077\text{x}10^{-5}$$

The above equation indicates that the approximation and quantization errors are influenced more by errors in the signal, $\hat{in}_5(\hat{t}_2)$, than those in the derivative, $\hat{in}'_5(\hat{t}_2)$. Combining Eq. 5C.30 with Eq. 5C.4 we see that the magnitude of the total error E_T is bounded by:

$$|E_T| = \left|e_{t_{nr}}\right| + \left|e_{a_{nr}}+e_{q_{nr}}\right| < 3.60747\text{x}10^{-6} + 2.48077\text{x}10^{-5} \qquad (5C.31)$$

$$\approx 2.84152\text{x}10^{-5}$$

We do stress while this bound is not far from the absolute limit imposed by the 16 bit quantization noise error, the result should be viewed as conservative. It is an *absolute worst case* limit (i.e., a 20kHz, full scale input combined with a special sequence of quantization errors on the input signal required to make the noise propagation effects large. Either of these scenarios is unlikely to occur in typical digital audio applications. Also, due to the excess bandwidth created by the oversampling, the errors which do occur are likely to be distributed over the entire band, $[0,\frac{1}{2}f_c=176.4kHz)$, with only a portion falling in the audio band, $[0,f_b=20kHz)$. (Computer simulations presented in Chapter Seven verify this effect.) A more accurate (but more difficult) statistical analysis would take these issues into account and derive a bound for the error power and more generally an expression for the PSD of the error for a given input.

Appendix 5D: Error Analysis for the Third Order Algorithm

In this appendix we present an error analysis for the third order algorithm introduced in Section 5.3.3. Since the development is identical in structure to that of Appendix 5C for the fifth order algorithm our analysis appears in an abbreviated form.

As in Appendix 5C the error inherent to the technique is:

$$\left|e_{t_m}\right| \leq \sim 3.60747\text{x}10^{-6} \tag{5D.1}$$

For the approximation error it is easily verified that:

$$\underset{t\in[\tau_{-1},\tau_2]}{Max}\left|\Psi_3(t)\right| \leq \sim 1.0 \quad \underset{t\in[\tau_{-1},\tau_2]}{Max}\left|\Psi'_3(t)\right| \leq 6.0 \tag{5D.2}$$

Hence,

$$|\hat{e}_3(t)| \leq \underset{t\in[\tau_{-1},\tau_2]}{Max}\left|\frac{\Psi_3(t)}{4!}f^{(4)}(t)\right| \leq \sim 3.35339\text{x}10^{-4} \tag{5D.3}$$

$$|\hat{e}_{\sim 3}(t)| \leq \underset{t\in[\tau_{-1},\tau_2]}{Max}\left|\frac{\Psi'_3(t)}{4!}f^{(4)}(t)\right| + \underset{t\in[\tau_{-1},\tau_2]}{Max}\left|\frac{\Psi_3(t)}{5!}f^{(5)}(t)\right| \tag{5D.4}$$

$$< \sim 2.01203\text{x}10^{-3} + \sim 2.38888\text{x}10^{-5} \simeq 2.03592\text{x}10^{-3}$$

where $\tau_i \quad i \in \{-1,0,1,2\}$ are numerically the same as defined after Eq. 5C.8 of Appendix 5C.

It has been shown by computer simulation that:

$$\underset{\substack{t\in[\tau_{-1},\tau_2)\\ |\varepsilon_i|\leq\varepsilon_{max},\ i\in\{-1,0,1,2\}}}{Max}\left|\hat{\varepsilon}_c(t)\right| \leq \sim 1.24435\text{x}10^{-5} \tag{5D.5}$$

$$\underset{\substack{t\in[\tau_{-1},\tau_2)\\ |\varepsilon_i|\leq\varepsilon_{max},\ i\in\{-1,0,1,2\}}}{Max}\left|\hat{\varepsilon}'_c(t)\right| \leq \sim 1.08086\text{x}10^{-4} \tag{5D.6}$$

Following the development from Appendix 5C, the above values give rise to:

$$Max\,|C| \simeq 9.66854\text{x}10^{-3} \tag{5D.7}$$

$$Max\,|E_C| \simeq 3.35339\text{x}10^{-4} + 1.24435\text{x}10^{-5} \simeq 3.47783\text{x}10^{-4} \tag{5D.8}$$

$$Max\,|D| \simeq 1.21669 \tag{5D.9}$$

$$Max\,|E_D| \simeq \frac{1}{1-\{(0.5)(0.35619)+2.03592\text{x}10^{-3}+1.08086\text{x}10^{-4}\}} \tag{5D.10}$$

$$-\frac{1}{1-(0.5)(0.35619)} \simeq 3.18212\text{x}10^{-3}$$

These finally imply that:

$$\left|e_{a_{nr}} + e_{q_{nr}}\right| < (3.47783\text{x}10^{-4})(1.21669) + (3.18212\text{x}10^{-3})(9.66854\text{x}10^{-3}) \qquad (5D.11)$$

$$+ (3.47783\text{x}10^{-4})(3.18212\text{x}10^{-3}) \approx 4.55017\text{x}^{-4}$$

and

$$|E_T| \approx 3.60747\text{x}10^{-6} + 4.55017\text{x}10^{-4} \approx 4.58625\text{x}10^{-4} \qquad (5D.12)$$

We see that the maximum error for the third order procedure is between that of the first order procedure and fifth order technique. While this figure may lead us to believe that 16 bit quality performance is not possible with the third order algorithm, we should bear in mind that $|E_T|$ is an absolute worst case error. Also it says nothing about the structure of the error. In fact, in Chapter Seven we see that the third order procedure performs quite well for many inputs, with much of the error residing outside the baseband.

Chapter Six
The Simulation

6.1 Introduction

A generalized block diagram of the systems we are interested in studying is shown in Fig. 6.1. In the first stage an interpolator raises the sampling rate of the input (obtained from some source such as a compact disc player) by a factor, L. Next, for PNPWM DACs, the pseudo natural cross point times are computed by one of the cross point algorithms described in Chapter Five. Of course, if the DAC is UPWM based the cross point deriver is simply bypassed. A noise shaping network of the form described in Chapter Four is used to reduce the wordlength of its input with negligible loss in baseband SNR. Its output drives a digital uniform sampling pulse width modulator of appropriate wordlength. If the system is for high power level digital-to-analogue conversion, then a power switching stage is used for class D amplification. For ordinary, low power level conversion the power switching stage is bypassed. Lastly, an analogue low pass filter is used to reveal the final analogue output waveform. (Of course, the choice of components for such a filter would depend on whether the converter was intended for power level or signal level conversion.)

The preceding four chapters have been devoted to the analysis of several of the individual blocks in Fig. 6.1. In spite of these theoretical considerations, we have decided to computer simulate the performance of various ONS/PWM DAC structures. This is for several reasons. First, the cross point deriver, the noise shaper, and the pulse width modulator are all *nonlinear* systems which have each been analyzed in isolation. Hence, it is difficult to know precisely how they will interact when connected together. Such knowledge is essential for making appropriate decisions on how to combine these blocks in order to maximize performance. Also the analysis of each stage has often been limited due to restrictive assumptions about the system input or about the system itself. In addition, there are other smaller effects (such as the extent to which errors in the cross point

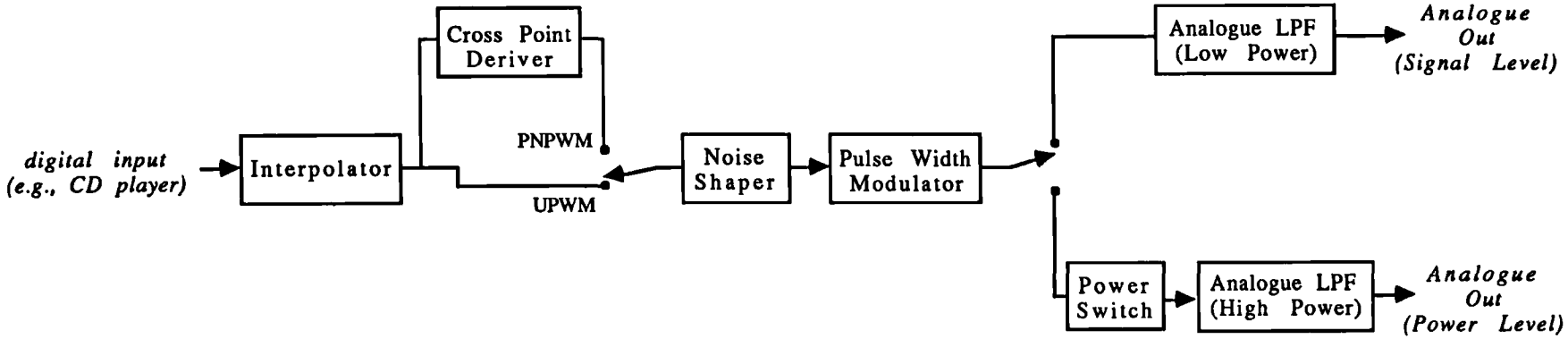


Fig. 6.1: Block Diagram of General ONS/PWM DAC

computation are signal correlated) which should be considered since they may detract from overall DAC quality. As a complete theoretical analysis is difficult, computer simulations represent a quicker and easier way to gain insight into the behaviour of these systems.

The simulation has been used with great success. Many basic theoretical results were verified, while in other instances previously unforeseen problems have been brought to light. The simulation is based on a collection of *C* language programs and shell scripts run on a Unix based workstation. A block diagram of its structure is shown in Fig. 6.2. Small modules of *C* code have been written which correspond to the individual stages of signal processing that would comprise a hardware realization of the DAC under simulation. In fact, the similarities between Figs. 6.1 and 6.2 are readily apparent. The modularity of the software makes the simulation flexible and intuitive to use. The behaviour of individual stages of signal processing can be tested easily under various conditions. It is also possible to rearrange or omit some of the modules in Fig. 6.2, allowing us to examine a variety of DACs. There are several additional supporting design and analysis tools.

This chapter describes how each stage of the simulation has been implemented. (The actual source code is provided on a 3½ inch floppy disc attached to the back cover of this thesis.) We will see that several of the modules correspond very closely to direct software implementations of the signal processing algorithms described in Chapters Three, Four, and Five. Other modules use special techniques devised to improve the efficiency of the simulation. Digital filter design and analysis procedures along with signal analysis techniques are also described. We conclude with brief comments on the limitations of the simulation.

6.2 Signal Generators

The signal generators produce various input test signals which can be used to evaluate the performance of the DACs we wish to simulate. They are also useful for examining other combinations of signal processing operations discussed in the previous chapters. In Fig. 6.3 we see that there are two independent signal generators. The ordinary "Signal Generator" is capable of producing signals which are a sum of P ($P \leq 10$) individual, discrete time sinusoids of user-defined amplitude, A_i, frequency, f_{v_i}, and phase, ϕ_i, $i \in \{0,1,\cdots,P-1\}$. The output is quantized to b bits. There is also full control over the sampling frequency, f_s. The signal is sent to the standard output in "chunks" of 256 sample values in a double precision, floating point, binary format. The total number of output samples is controlled by specifying R, the desired number of output chunks. We note that the output signal file contains the sampling frequency as its first entry. The "Cross Point

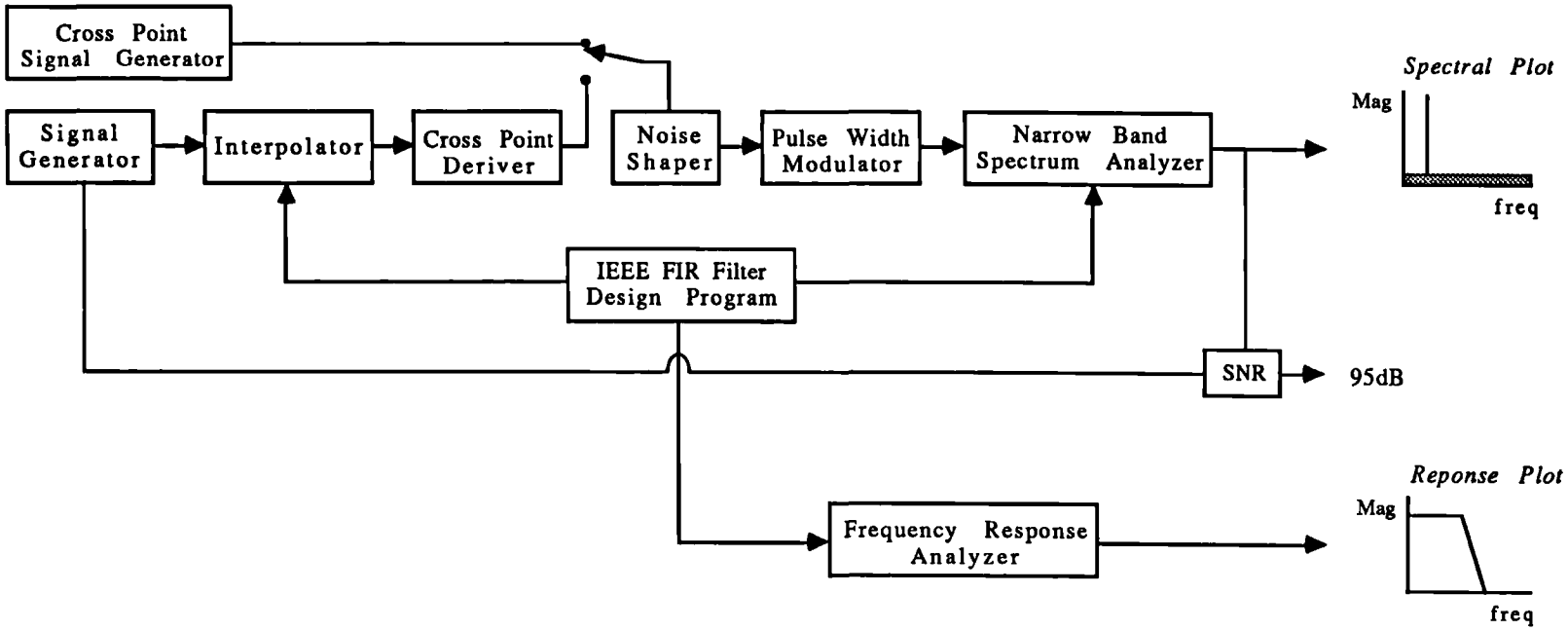


Fig. 6.2: Block Diagram of the Simulator and Support Tools

Signal Generator" automatically outputs the PNPWM cross points times for the sum-of-sinusoids input specified by the parameters above.* The cross point times are derived from a crude binary search rootfinding algorithm. (See [At89].) The program is given the input signal parameters and explicitly evaluates the input signal (via calls to the *sine* function) in its search for the time at which the input crosses the comparison waveform. The cross point time will possess b bit accuracy. As such it is particularly useful as a benchmark for comparisons with the performance of approximate algorithms (such as those of Chapter Five). However, for a given wordlength we note that the baseband quantization noise power of the Cross Point Signal Generator output is always lower than that of the more realistic interpolator/cross point deriver combination of Fig. 6.1. This is because the output of the former essentially consists of quantized samples of an analogue signal taken at the high (oversampled) rate rather than quantized samples taken at the Nyquist rate, interpolated up, and "cross point derived."

6.3 Interpolator

The interpolator is used to raise the sampling rate of a signal by a factor, L. As explained in Chapter Two, interpolation is essentially a digital filtering operation which can be realized more efficiently in an I-stage multistage implementation. This is shown in Fig. 6.4 for an input signal, $x[n]$, of length J samples and an output signal, $y[m]$, of length LJ. The input and output data format between each stage is the same as the output format described in the previous section.

As explained in Chapter Two, the number of stages, I, as well as the interpolation factor for each stage, L_i, ($\prod_{i=0}^{I-1} L_i = L$), are chosen to reduce overall computational complexity. The ith stage uses a linear phase FIR filter of length, N_i, designed with a standard algorithm described later in this chapter. The implementation of each stage of the interpolator is based on the efficient procedures described in Chapter Three.

* In each of the following sections we denote the input to and output from each module as $x[n]$ and $y[n]$, respectively.

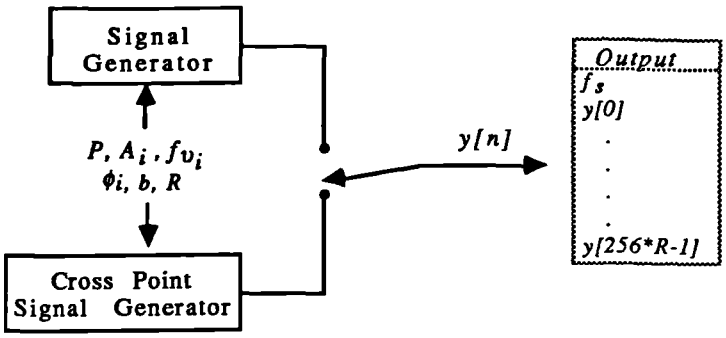


Fig. 6.3: Signal Generators

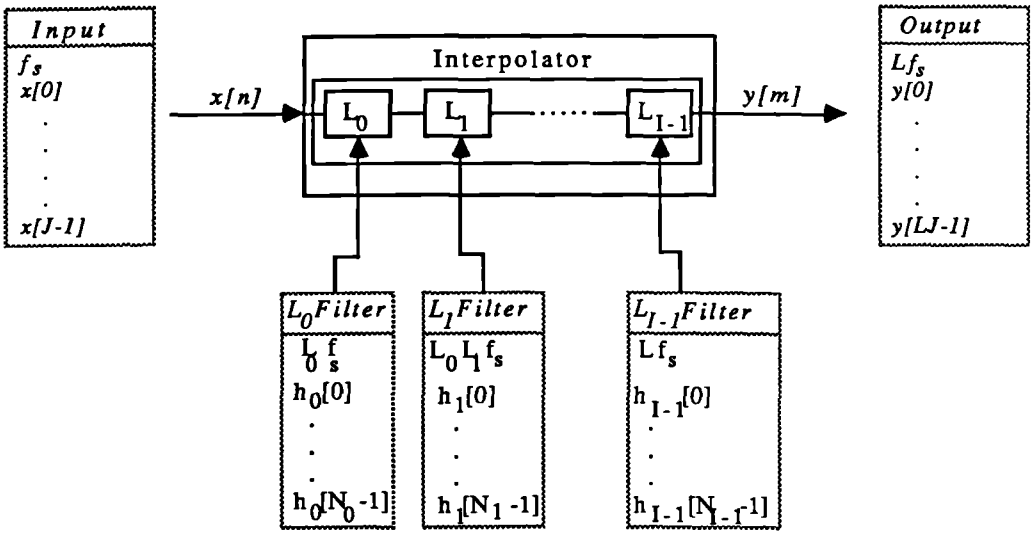


Fig. 6.4: Interpolator

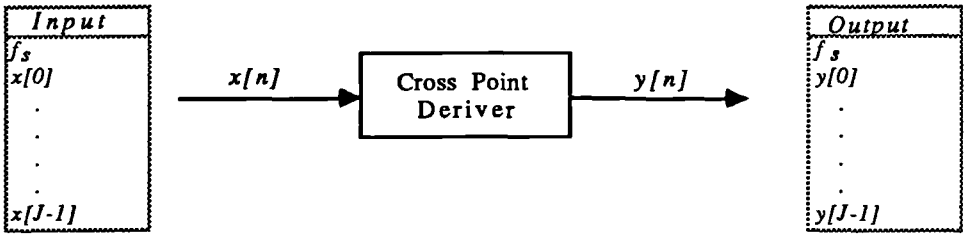


Fig. 6.5: Cross Point Deriver (Dedicated)

6.4 Cross Point Deriver

In Chapter Five we described in detail three cross point derivation algorithms. While all are based on polynomial signal approximation and rootfinding techniques, variations in the specific procedures are such that they are most easily realized as separate, dedicated programs. In this way the appropriate cross point deriver is chosen, and no parameters need to be passed to the routines. This is shown in Fig. 6.5.

6.5 Noise Shaper

As shown in Chapter Four, noise shaping uses a combination of oversampling, coarse quantization and error feedback techniques to accurately represent a high resolution signal with smaller wordlength than normally possible. The quantizer and limiter are fully specified by b and b', the input accuracy and output wordlength in bits, respectively. The limiter clips the input to the quantizer to between -2^{b-1} and $2^{b-1}-1$.

The noise shaper is designed for use with an Nth order FIR feedback filter, $H(z)$. Recall from Chapter Four that the output of this filter is always delayed by a least one sample which implies that $h[0] = 0$. This is shown in Fig. 6.6.

6.6 Pulse Width Modulator

The digital uniform sampling pulse width modulator converts its digital input into a sequence of high frequency pulses of varying width. The duration of these pulses is quantized in time with the same resolution that the modulator's input is quantized in amplitude. The total number of possible output pulse widths is controlled by setting upper and lower bounds for the modulator input, denoted by A_u and A_l, respectively. This is done in preference to simply supplying the modulator with the input signal wordlength because the above bounds can be set somewhat larger in magnitude than theoretically necessary to incorporate user-defined guard band regions. It is also possible to select one of several UPWM modulation types. This is shown in Fig. 6.7.

In a hardware implementation the output of the modulator is, of course, a continuous time waveform. However, in the simulation we represent this signal as a sequence of discrete time "highs" (i.e., +1's) and "lows" (i.e., -1's). The "effective sampling rate" at the output of the modulator is numerically equivalent to, f_{clk}, the modulator clock speed

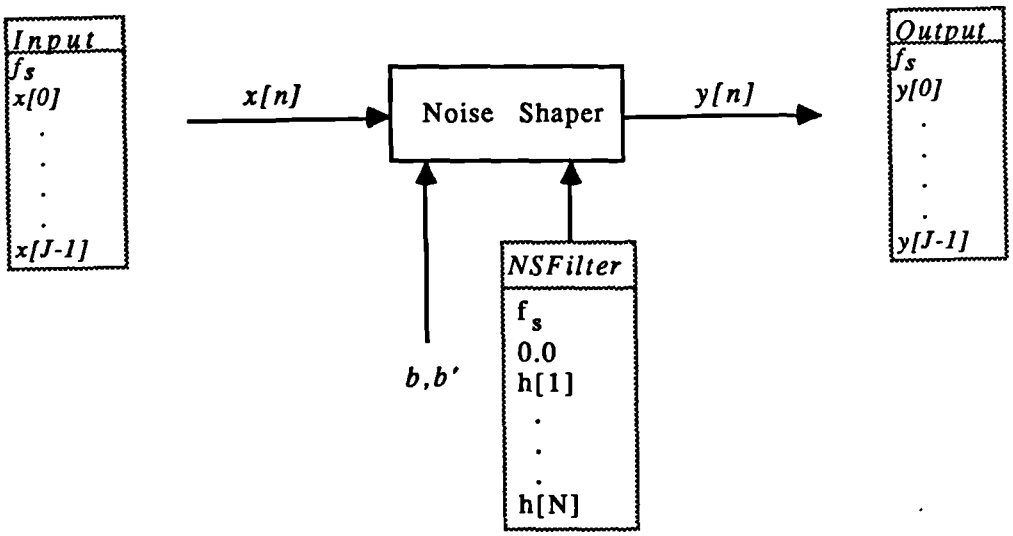


Fig. 6.6: Noise Shaper

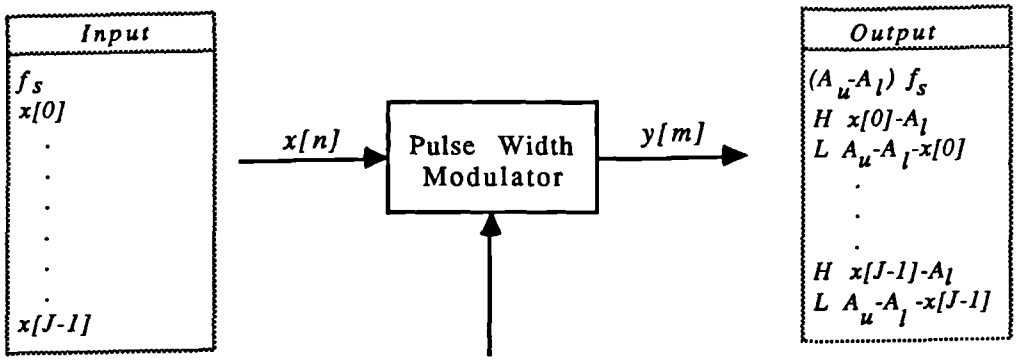


Fig. 6.7: Pulse Width Modulator (Trailing Edge Output)

(which is often orders of magnitude larger than the sampling rate of the original DAC input). To reduce the size of the output file the sequences are represented in a run length coded (RLC) format. The first line in the output file contains the effective sampling rate. Subsequent lines contain an "H" or an "L" to indicate whether the output level is high or low followed by the number of quantized time intervals associated with this level. The particular RLC format along with the effective sampling rate depend on the specific modulation type chosen. This dependence and the actual rules by which the PWM pulses are generated for each modulation type are shown in Table 6.1.

Table 6.1 PWM Simulation Output File Formats

Modulation Type	f_{clk}	RLC Format (for a single output pulse)
Trailing Edge	$(A_u-A_l+1)f_s$	$H\ x[n]-A_l$ $L\ A_u-x[n]+1$
Symmetric	$2(A_u-A_l+1)f_s$	$L\ A_u+1-x[n]$ $H\ 2(x[n]-A_l)$ $L\ A_u+1-x[n]$
Asymmetric	$(A_u-A_l+1)f_s$	$L\ \frac{(A_u-A_l+1)}{2}-(x[n]-A_l)+\frac{(x[n]-A_l)}{2}$ $H\ x[n]-A_l$ $L\ \frac{(A_u-A_l+1)}{2}-\frac{(x[n]-A_l)}{2}$
Two Sample Consecutive	$2(A_u-A_l+1)f_s$	$L\ A_u+1-x[n]$ $H\ x[n]+x[n+1]-2A_l$ $L\ A_u+1-x[n+1]$

In the table we assume that A_u-A_l+1 is even. Also, divisions by two are "integeric" divisions (e.g., $\frac{3}{2}=1$). The asymmetric format shown is for so-called "fixed asymmetric" modulation. When $x[n]-A_l$ is odd, the asymmetry is such that the duration of the first "low" time interval is one time quantum shorter than that of the second "low" interval. It is also possible to generate an "alternate asymmetric" modulation type where for $x[n]-A_l$ odd the asymmetry alternates between the leading and trailing edges of the pulse (i.e., the first and third lines of the asymmetric RLC format in Table 6.1 alternate for odd-valued pulse durations).

6.7 Narrow Band Spectrum Analyzer

The narrow band spectrum analyzer is used to reveal the baseband frequency domain characteristics of a digital signal with a sampling rate, f_s, well in excess of twice the bandwidth of interest, $2f_b$, ($f_s \gg 2f_b$). The analyzer is comprised of a decimation section and a spectral estimation section. This approach, shown in Fig. 6.8, is well established for computationally, efficient narrow band spectral estimation [Li78].

6.7.1 Decimator

The decimator is used to lower the sampling rate of the input by an integer factor, M, to a new rate near or equal to $2f_b$. As was the case with interpolation, we have seen in Chapter Three that decimation is largely a digital filtering procedure which can be implemented in an efficient I-stage multistage structure. (In general the value of I in this section is not the same as that of Section 6.3.) The first stage of decimation can accept data either in the "regular" signal format described in Section 6.2 or, when sample rate reduction of the sampled PWM waveform is required, in the RLC format of Section 6.6. In the latter, very large decimation factors are common. In Chapter Four we have seen effective sampling frequencies in excess of 90MHz, implying decimation factors in excess of M=2048. In such instances the first stage can be realized with a very computationally efficient comb structure as described in Chapter Three. This is shown in Fig. 6.8 where the only parameter to be specified is, M_0, the decimation factor for the first stage. Subsequent stages of decimation are implemented using the efficient procedures described in Chapter Two. The filters used in these stages of decimation are of lengths, N_i $i \in \{1,2,\cdots,I-1\}$, which, in general, are different from the N_i associated with the interpolator.

6.7.2 Spectral Estimator

The spectral estimation section uses Welch's modified periodogram technique to obtain an estimate of the power spectrum of the output of the decimator (or any other signal processing module in Fig. 6.2). This method is well established as a simple and computationally efficient means of obtaining reasonable spectral estimates in a wide variety of applications where the signal can be modeled as a wide sense stationary random process.

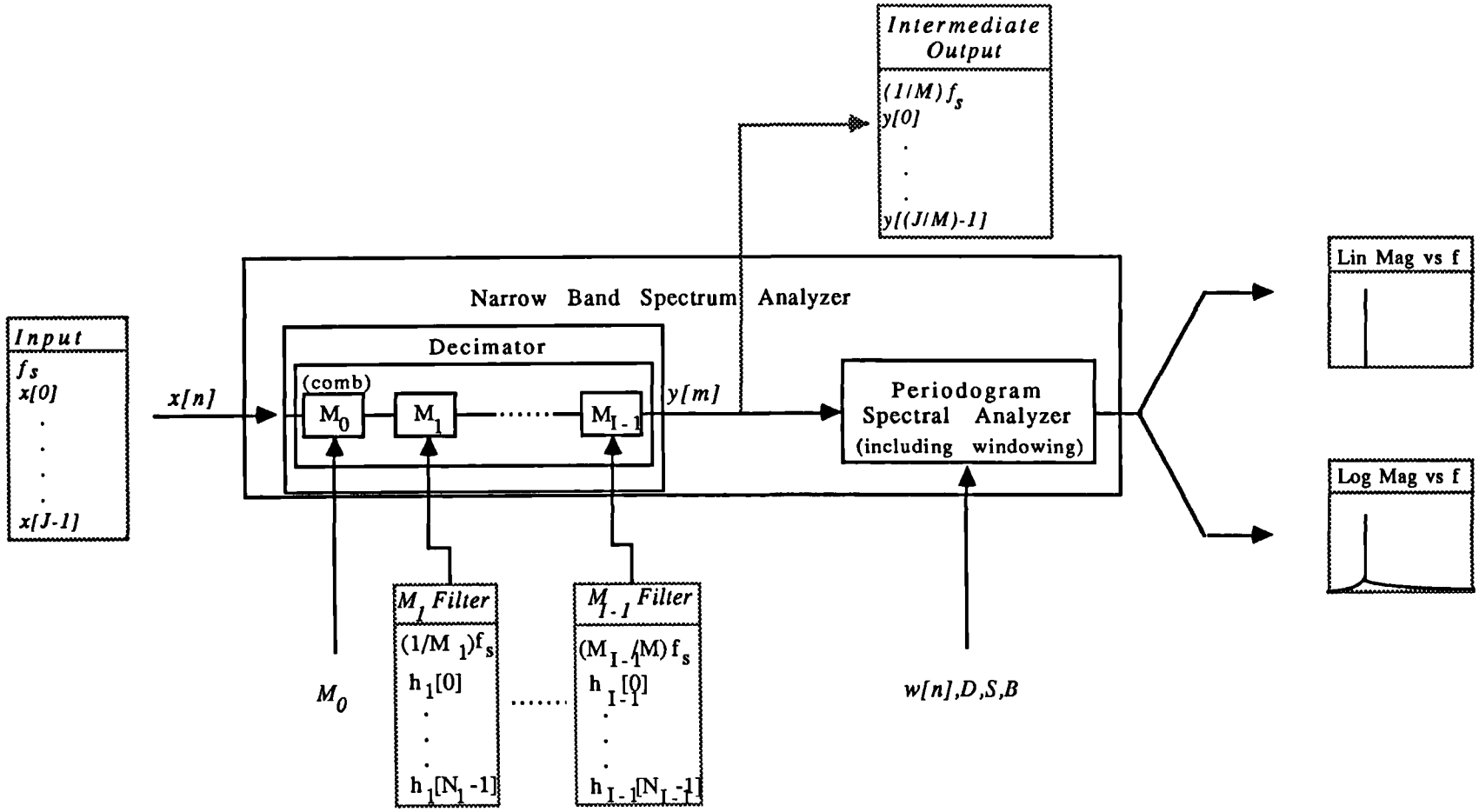


Fig. 6.8: Narrow Band Spectrum Analyzer

Here we give just a very brief description. More details can be found in [Pr89, Op89].

We assume that the input to the analyzer is a sequence, $y[n]$, of length C. First, a window of length D, $w[n]$ $n \in \{0,1,\cdots,D-1\}$, is used to partition $y[n]$ into K possibly overlapping blocks each of length, D:

$$v_b[n] = y[bB+n]w[n] \qquad b \in \{0,1,\cdots,K-1\},\ n \in \{0,1,\cdots,D-1\} \tag{6.1}$$

K is the largest integer such that $(K-1)B+D \leq C$. If $B<D$ then each block (except the first) overlaps the previous block by $D-B$ samples. The spectral estimator consists of an average of the squared magnitudes of the Fourier Transforms of the individual blocks. The averaging as well as the overlapping helps to reduce the variance of the estimate [Op89].

In particular, each block has a Discrete Time Fourier Transform (DTFT) given by:

$$V_b(e^{j\omega}) = \sum_{n=0}^{D-1} v_b[n]e^{-j\omega n} \qquad b \in \{0,1,\cdots,K-1\} \tag{6.2}$$

However, as the technique uses the Discrete Fourier Transform (DFT) only the discrete frequencies:

$$\omega = \omega_k = \frac{2\pi k}{D} \qquad k \in \{0,1,\cdots,D-1\} \tag{6.3}$$

are considered. (The sampling frequency is normalized to f_s=1.) Specifically, the Welch modified periodogram estimator is given by:

$$\hat{S}_y(\omega_k) = \frac{1}{K}\sum_{b=0}^{K-1}\frac{1}{DU}|V_b(e^{j\omega_k})|^2 \qquad k \in \{0,1,\cdots,D-1\} \tag{6.4}$$

where proper scaling is ensured with [Op89]:

$$U = \frac{1}{D}\sum_{n=0}^{D-1}(w[n])^2 \tag{6.5}$$

The specific choice of window can greatly influence the appearance of the resulting power spectrum estimate. Care must be taken in choosing an appropriate window. In the spectrum analyzer three windows are available: the rectangular window, the Hanning window, and a window which we call the "Nutall window" [Nu81]. Their respective equations are given in Table 6.2. Although the full analysis is beyond the scope of this chapter the frequency domain effect of a window is given by:

$$E[\hat{S}_y(\omega)] = \frac{1}{2\pi DU}\int_{-\pi}^{\pi} S_y(\theta)W(\omega-\theta)d\theta \tag{6.6}$$

Table 6.2 Comparison of DFT Windows		
Window Type	$w[n] \quad n \in \{0,1,\cdots N\}$	a_k
Rectangular	$w[n] = a_0$	$a_0 = 1.0$
Hanning	$w[n] = \sum_{k=0}^{1} a_k \cos\left[\frac{2\pi nk}{D-1}\right]$	$a_0 = 0.5$ $a_1 = -0.5$
Nutall	$w[n] = \sum_{k=0}^{3} a_k \cos\left[\frac{2\pi k}{D-1}\left[n-\frac{D}{2}\right]\right]$	$a_0 = 0.338946$ $a_1 = 0.481973$ $a_2 = 0.161054$ $a_3 = 0.018027$

with

$$W(\omega) = \left| \sum_{n=0}^{D-1} w[n] e^{-j\omega n} \right|^2 \tag{6.7}$$

where $E[\cdot]$ denotes the expectation operator and $S_y(\omega)$ is the power spectral density of the wide sense stationary random process, $Y[n]$, of which $y[n] \quad n \in \{0,1,\cdots,C-1\}$ is considered a truncated realization. Full details can be found in [Op89]. From Eqs. 6.6 and 6.9 we see that the effect of the window is that of a frequency domain convolution. Plots of the magnitude of the DFT are shown in Figs. 6.9a-c, respectively.* In the plots it is important to note the width of the main lobe at DC along with the height of the side lobes at other frequencies. It can be seen from Eqs. 6.7 and 6.8 and Fig. 6.9 that large main lobe widths result in a "loss of frequency resolution." As a result of the convolution, individual spectral peaks become broadened in a so-called "frequency smearing" effect. This makes it difficult to resolve two peaks which are very close in frequency. On the other hand, high sidelobe levels result in a so-called "spectral leakage" effect causing the power content at one frequency to "leak" into other (possibly far away) frequency regions. For our three windows we see that a tradeoff is possible between main lobe width and side lobe height. The rectangular window has a narrow main lobe width and large side lobes while the opposite is true of the Nutall window. The Hanning window is in between.

As seen in Fig. 6.8 the DFT length, D, the window type, the "overlap factor," B, and a "skip factor," S, are all passed to the spectrum analyzer. This last quantity allows the analyzer to "skip over" (i.e., not consider) the first S samples of the incoming sequence. This is useful when we are analyzing the output of a digital filter where the first few

* For purposes of illustration D has been set artificially low to D=256. In practice D will be larger. (See Chapter Seven.)

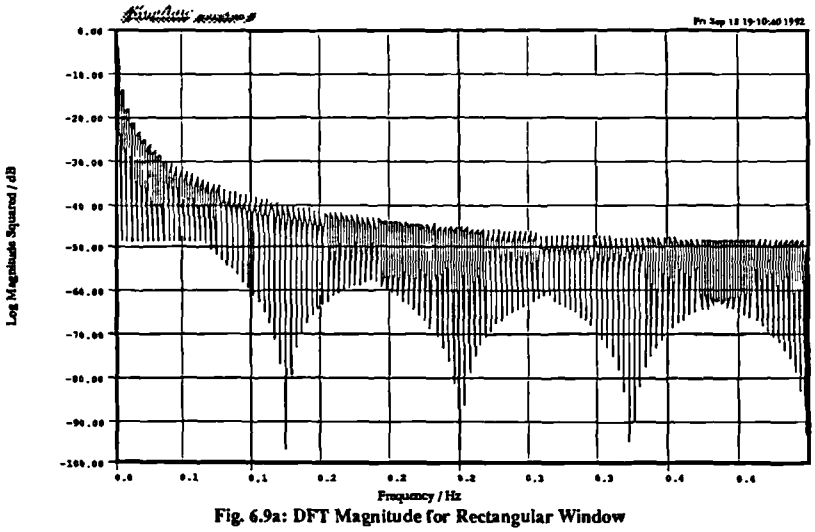


Fig. 6.9a: DFT Magnitude for Rectangular Window

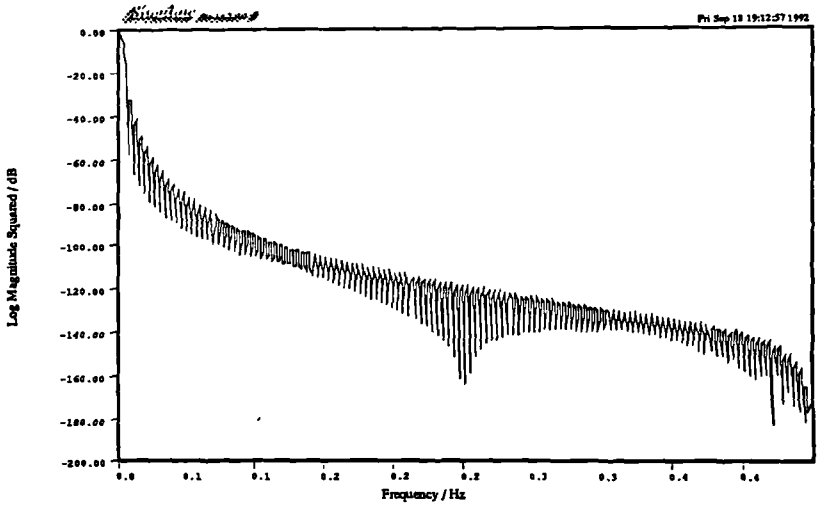


Fig. 6.9b: DFT Magnitude for Hamming Window

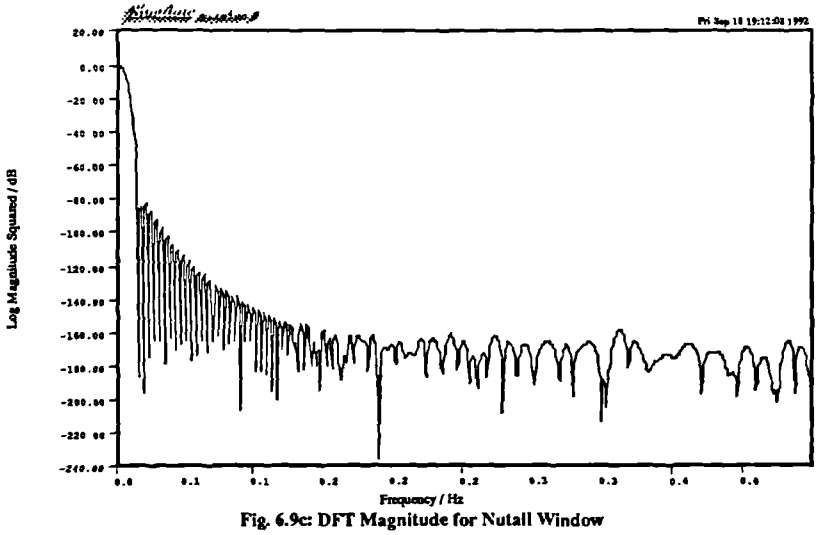


Fig. 6.9c: DFT Magnitude for Nutall Window

samples correspond to the "startup" transient. The plots of the linear magnitude and logarithmic magnitude of the spectral estimate are available.* Hard copies are also easily obtained.

6.7.3 SNR Calculation

System quality can also be assessed in terms of signal-to-noise ratio (SNR). In the time domain this is computed as:

$$SNR = \frac{\frac{1}{M-1}\sum_{m=0}^{M-1}(x[n])^2}{\frac{1}{M-1}\sum_{m=0}^{M-1}(e[n])^2} = \frac{\frac{1}{M-1}\sum_{m=0}^{M-1}(x[n])^2}{\frac{1}{M-1}\sum_{m=0}^{M-1}(x[n]-y[n])^2} \tag{6.8}$$

Where $x[n]$ and $y[n]$ are the system input and output, respectively, and $e[n]=x[n]-y[n]$ is the error signal. As M approaches infinity the numerator and denominator of the above equation will approach the average power of the input and the error signals, respectively. The above equation is over-simplified however, because delay is often introduced in the system under test. There are also transient times which can affect the SNR measurement. So in practice it is necessary to properly align the signals in time and wait until the system transients have decayed before making measurements on the basis of Eq. 6.8.

6.8 Digital Filter Design and Analysis Tools

We have seen that digital filtering is used extensively in many of the signal processing networks described in this thesis. Specifically, interpolation and noise shaping feedback filters are used inside the actual DACs (and are simulated as such) while decimation filters are used only inside the narrow band spectrum analyzer of the simulation. A special design procedure for noise shaping feedback filters was mentioned in Chapter Four. For the interpolation and decimation filter the equiripple design algorithm mentioned in Chapter Three is used. This procedure produces linear phase FIR filters with minimized maximum weighted deviation from the ideal response for a given filter length, N. The important specifications for such (low pass) filters are the passband and stopband frequencies, the passband and stopband ripples, and the sampling frequency. These parameters are denoted

* The graphics routines and the spectrum analyzer were written by Allan Paul, a colleague at King's.

as f_1, f_2, δ_1, δ_2, and f_s, respectively.* From these an estimate of the minimum required filter length, N, is derived.

The actual code for the equiripple design algorithm was obtained from a widely available collection of digital signal processing computer programs [Mc79]. It is used with a well known filter length estimation formula [Va87]. The five parameters listed above are supplied by the user and a suggested filter length is derived by the length estimation program. This suggestion can be used or a new filter length, N', can be specified. The structure of the design procedure is shown in Fig. 6.10. It is also possible to design low order multi-band filters along with the half-band filters described in Chapter Two.

In Fig. 6.10 we also see a frequency response analyzer which can be used for any FIR filter. The frequency response of a filter of length, N, is estimated by computing an $\hat{N}$ point DFT of a zero-padded version of $h[n]$ ($n \in \{0,1,\cdots,N-1\}$ $N<\hat{N}$), the impulse response of the filter. Specifically we compute:

$$\hat{H}\left(e^{j2\pi f_s k/\hat{N}}\right) = \sum_{n=0}^{\hat{N}-1} \hat{h}[n] e^{-j2\pi f_s kn/\hat{N}} \tag{6.9}$$

where $\hat{h}[n]$, the zero-padded impulse response, is given by:

$$\hat{h}[n] = \begin{cases} h[n] & n \in \{0,1,\cdots,N-1\} \\ 0 & n \in \{N,N+1,\cdots,\hat{N}-1\} \end{cases} \tag{6.10}$$

A graphics package is used to display plots of the linear magnitude, logarithmic magnitude, and the phase responses. As before, hard copies are available. $\hat{N}$ is internally selected as 1024. We note that, of course, the zero padding does not change the magnitude response of the filter but simply gives us a means to obtain a more detailed picture of its frequency response.

6.9 A Note on the Scope of the Simulation

We now briefly mention some of the limitations of the simulation. First, the nonidealities of the output stage such as pulse edge rounding, pulse edge jitter, and power supply instability are not simulated. (See Chapter Two.) These complicated effects are better understood from actual hardware measurements rather than software simulation. Research is currently underway [Hi92d]. Also, the effects of any spurious radio frequency

* For convenience there has been a change in notation for the filter design parameters from those given in Chapter Three.

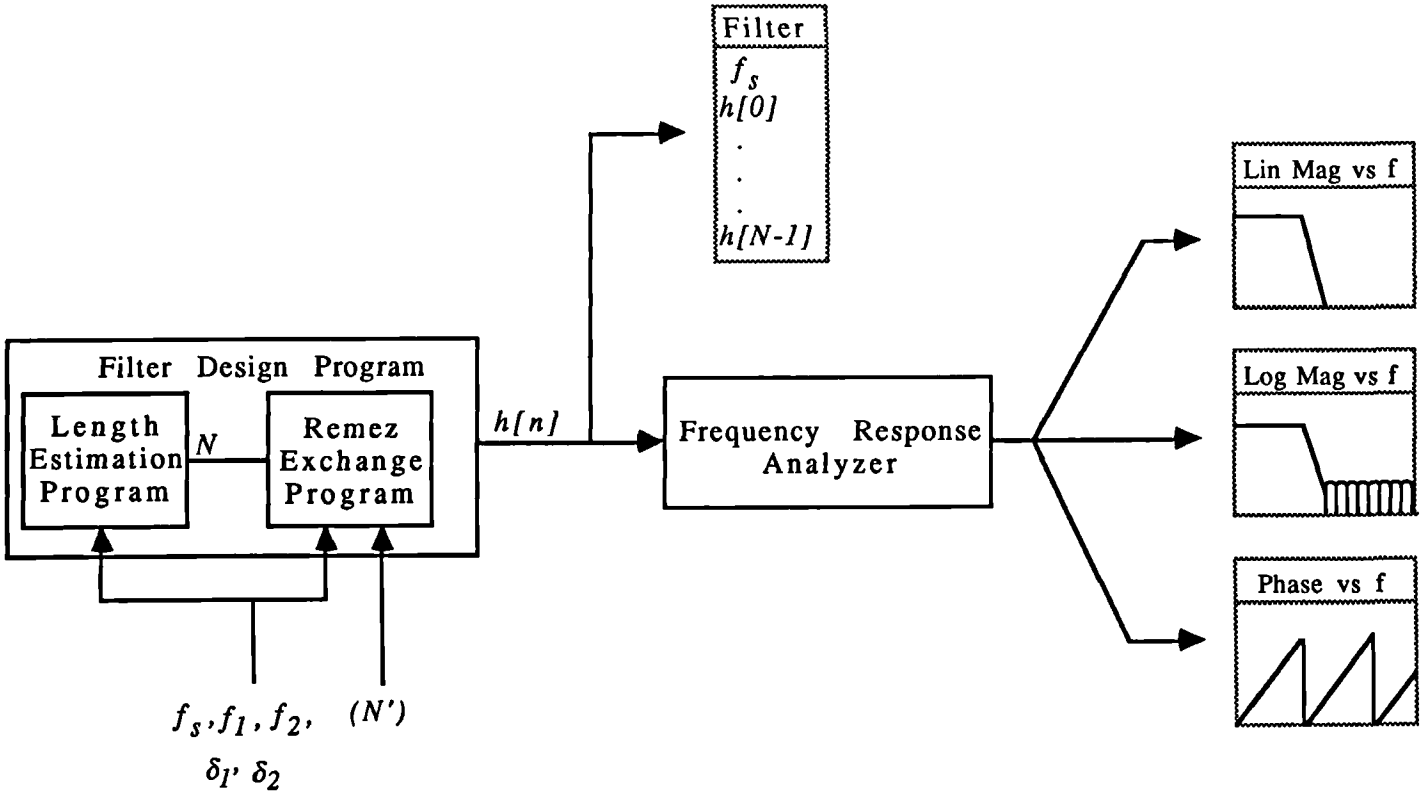


Fig. 6.10: Digital Filter Design and Analysis Tools

electro-magnetic interference emanating from a power level DAC with high frequency switching waveforms is not considered. Again this is best understood by conducting experiments with a hardware prototype. Lastly, the analogue low pass filters of Fig. 6.1 are in effect digitally simulated in the decimation procedure.

6.10 Summary

In this chapter the basic structure of our simulation of ONS/PWM DACs and their constituent parts has been presented. The programs which simulate parts of the DAC and other routines which provide design and analysis facilities were described. User control over each module is exercised through the numerical specification of parameter values. The modular structure of the simulation makes it an intuitive, flexible tool for analyzing many different DAC configurations or smaller associated signal processing networks. If desired in the future, the simulation can be expanded easily with the creation of additional modules to simulate new systems with additional or alternative stages of signal processing. (See [Pa92a].) Lastly, the limitations of the simulation were discussed. In the next chapter we present an extensive set of performance results based on the simulation.

Chapter Seven
Investigations and Results

7.1 Introduction

In Chapters Two through Five we introduced the various building blocks used to construct an ONS/PWM based DAC. Aside from the modulator itself, these blocks are intended either to improve the linearity of the DAC or to make the DAC more practical to realize in hardware. Specifically, the sample rate increase stage is used to reduce the harmonic (UPWM) and inharmonic (UPWM and NPWM) baseband distortion arising from the PWM modulation processes. (It also provides the excess bandwidth required by the noise shaper. See below.) As indicated in Chapter Two, even with relatively high oversampling ratios, it is the presence of the harmonic (as opposed to inharmonic) distortion which represents the main PWM performance limitation (discounting practical problems). For this reason the cross point deriver is used to digitally mimic the sampling process associated with the harmonic distortion free, continuous time NPWM modulator. Lastly, the noise shaper uses a combination of oversampling, coarse quantization, and error feedback filtering to reduce the wordlength of a high resolution signal with negligible loss in the overall baseband SNR. This enables very large reductions in the associated internal modulator clock speed thus making PWM more practical to implement. This chapter brings these building blocks together by presenting results from computer simulations of several relevant PWM DAC systems which are comprised of various combinations of the subsystems mentioned above.

The structure of this chapter broadly follows that of the thesis as a whole. We begin by examining the performance of the basic UPWM modulation types with single tone and twin tone inputs for a variety of pulse repetition frequencies. This is followed by a series of tests on similar UPWM systems but with some form of oversampled noise shaping preceding the actual modulator. Results are then presented for PWM based DACs using both ONS and PNPWM cross point derivation. Due to the large number and variety of

issues discussed in this chapter we have chosen to include short "overview" sub-sections at the end of each major section with a brief summary at the end of the chapter.

The PWM modulation types we consider in this chapter are denoted by the letters *a* through *p* as shown in Table 7.1. (Those modulation types which have not already been introduced will be described as necessary in the course of this chapter.)

Table 7.1: Key for Digital PWM Modulation Types

Symbol	Modulation Type
a	UPWM single sided trailing edge
b	UPWM double sided symmetric
c	UPWM double sided asymmetric (fixed odd sample)
d	UPWM double sided asymmetric (alternate odd sample)
e	UPWM double sided asymmetric (two sample consecutive)
f	PNPWM single sided trailing edge ("perfect")
g	PNPWM single sided trailing edge ("perfect" Newton iteration)
h	PNPWM single sided trailing edge (1st order)
i	PNPWM single sided trailing edge (3rd order)
j	PNPWM single sided trailing edge (5th order)
k	PNPWM single sided trailing edge (3rd order 1 Xpt per polynomial)
l	PNPWM single sided trailing edge (5th order 1 Xpt per polynomial)
m	PNPWM single sided trailing edge (3rd order perfect derivative)
n	PNPWM single sided trailing edge (3rd order perfect signal)
o	PNPWM single sided trailing edge (5th order perfect derivative)
p	PNPWM single sided trailing edge (5th order perfect signal)

7.2 UPWM Based DACs

In this section we present results from computer simulations of conventional UPWM based DAC systems such as the one shown in Fig. 7.1. In particular, we show for several tone inputs the effect of varying the pulse repetition frequency and the modulation depth on the levels of baseband harmonic and foldback distortion terms. These results are shown

to agree closely with those predicted theoretically in Chapter Two. We also present twin tone test results which indicate the presence of *intermodulation* distortion terms and show how these change with pulse repetition frequency.

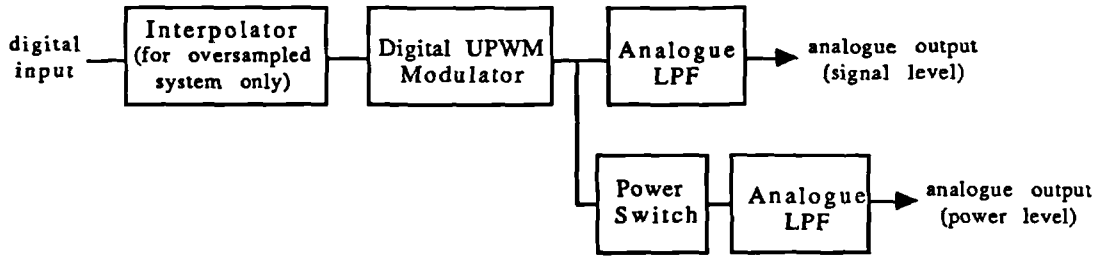


Fig. 7.1: Basic UPWM DAC (with overampling)

We begin by presenting a series of wide band plots of the output tone spectra for the five UPWM modulation types.* They are shown in Fig. 7.2a-e, for modulation types *a* through *e* respectively. In each case the pulse repetition frequency, f_c, is 44.1kHz while the input consists of a 16 bit sinewave of frequency $f_v = 2.001kHz$ with a peak amplitude equal to 80% of full scale (i.e., M=0.8). In all five plots we note the presence of the input tone and harmonics in addition to higher frequency spectral content centered about multiples of the carrier frequency. It is seen that the trailing edge plot has by far the largest amount of high frequency content. Also, note that the spectra of the three double sided modulation types look very similar. (We will observe subtle differences between these later.) In addition, we see that the two sample consecutive modulation type exhibits only odd harmonics of the input and only alternate sideband terms about the carrier and its multiples. The baseband performance is now examined in more detail.

Consider the effect of varying the pulse repetition frequency on the levels of harmonic distortion associated with the five UPWM modulation types. Increasing f_c implies the need for a corresponding increase in f_s, the sampling rate of the signal applied to the

* In most cases the spectral plots in this chapter are based on averaging several overlapped 4096 point DFTs with the data pre-windowed by the "Nutall window" described in Chapter Six. The plots are normalized such that the 0dB level coincides with the peak value on the plot.

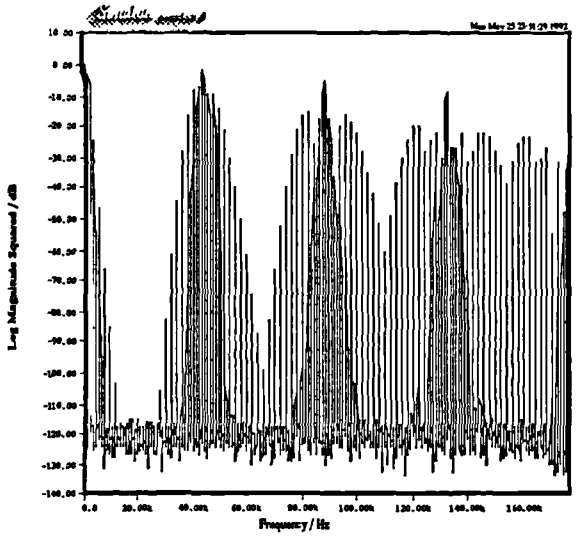


Fig. 7.2a: Wideband Spectrum (UPWM a)

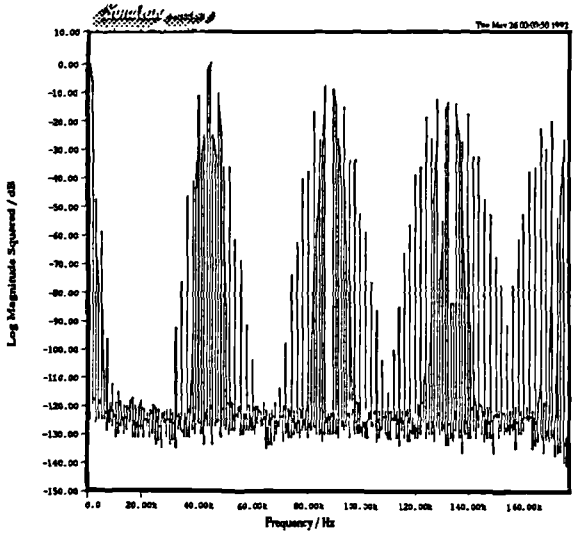


Fig. 7.2b: Wideband Spectrum (UPWM b)

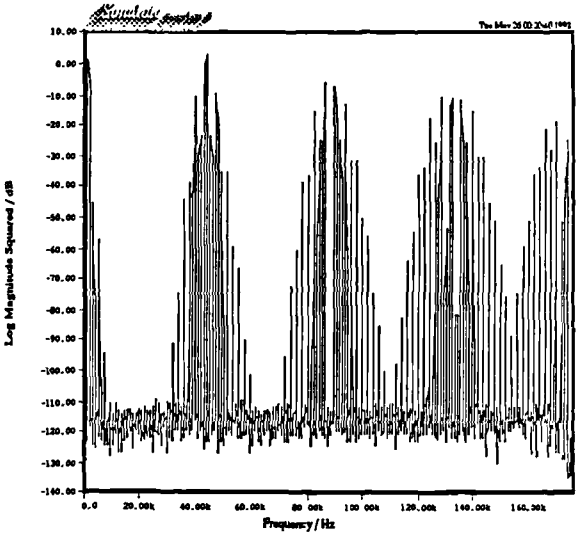


Fig. 7.2c: Wideband Spectrum (UPWM c)

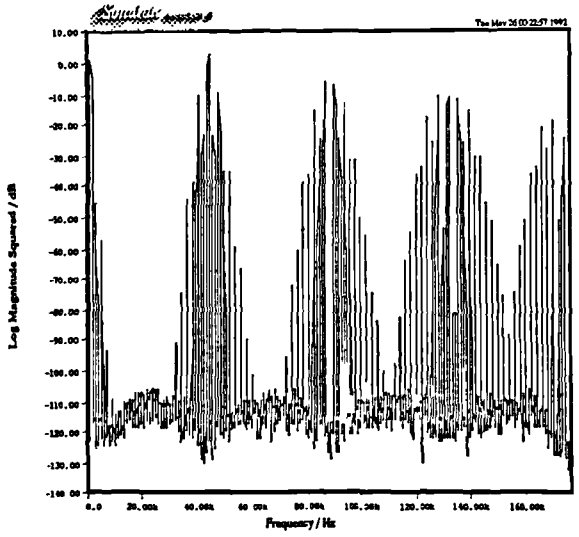


Fig. 7.2d: Wideband Spectrum (UPWM d)

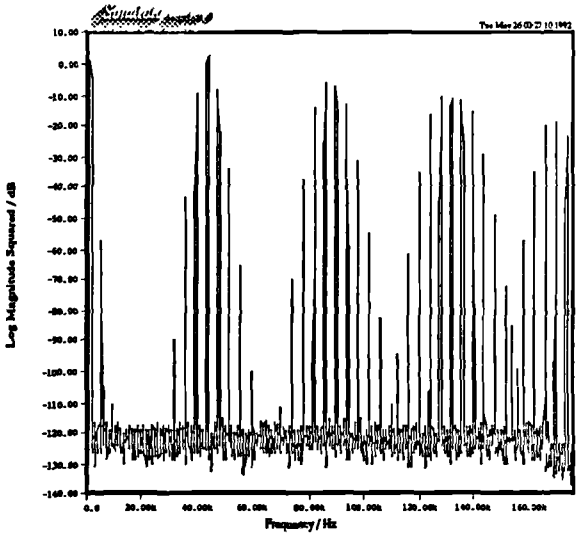


Fig. 7.2e: Wideband Spectrum (UPWM e)

modulator's input. In practice this is achieved by placing an interpolator directly before the modulator as was shown in Fig. 7.1. However, the results presented in this subsection are generated more simply by using the sinewave generator to produce data directly at the higher rate—eliminating the need for interpolation.†

We start with a 16 bit, 6.001kHz input signal with M=0.8. The pulse repetition frequency is set to f_c=44.1kHz. Fig. 7.3a-e shows the baseband output spectra for the five modulation types, a through e, respectively. The trailing edge spectrum shown in Fig. 7.3a indicates the presence of large distortion components at the second (2·6.001kHz=12.002kHz) and third (3·6.001kHz=18.003kHz) harmonics of the input tone with the former larger in magnitude than the latter. There are also inharmonic "foldback" terms arising from sideband components at the fourth, fifth, and sixth multiples of the input frequency about the pulse repetition frequency (i.e., |1·44.1kHz−4·6.001kHz|=20.096kHz, which is, strictly speaking, outside the 20kHz baseband, |1·44.1kHz−5·6.001kHz|=14.095kHz, and |1·44.1kHz−6·6.001kHz=8.094kHz|, respectively). The three double sided spectra in Fig. 7.3b-d are all similar. While they also possess distortion components at many of the same frequencies as in Fig. 7.3a, we see that each distortion component is smaller than that of the corresponding trailing edge component. Lastly, Fig. 7.3e shows the tone spectrum for two sample consecutive modulation. Here we immediately note the absence of the second harmonic distortion component as well as the foldback distortion component at the fifth multiple of the input frequency back from the pulse repetition frequency. We also see that the remaining third harmonic distortion term is quite similar in size to those of the double sided spectra while the two sample consecutive 20.096kHz foldback term is slightly larger than those of the double sided spectra.

Fig. 7.4a-e shows the results of the same tests but with a pulse repetition frequency of f_c=352.8kHz. In all five plots we notice that the increased pulse repetition frequency has resulted in a large reduction in distortion as compared with the corresponding f_c=44.1kHz set of Fig. 7.3. As before the level of distortion at the second harmonic is larger in size than that of the third (except for two sample consecutive modulation). Also, we see that for all five modulation types the foldback distortion has disappeared beneath the 16 bit noise floor and is therefore negligible.

---

† It should be noted that this approach results in a reduction in the total noise power in the output signal (3dB per doubling of sampling rate). This effect would not manifest itself if we had explicitly used sample rate increase devices between the non-oversampled (i.e., Nyquist) sampled input and the modulator. However, the levels of distortion (which are what we are interested in now) will remain virtually the same in either case. Actual interpolators are used explicitly in subsequent sections where we consider more realistic systems.

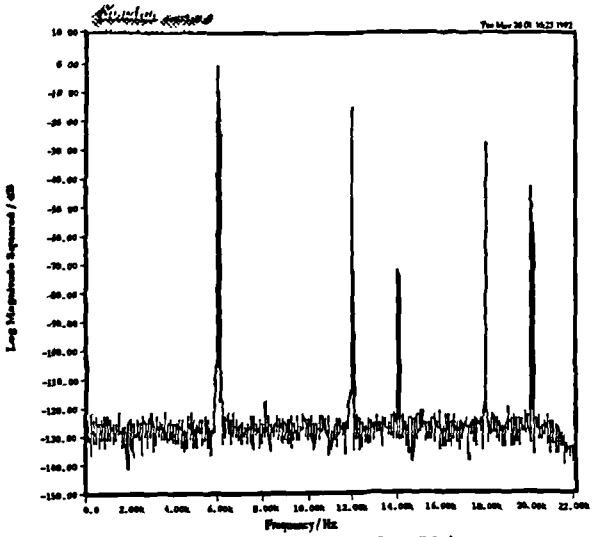


Fig. 7.3a: Baseband Spectrum (UPWM a)

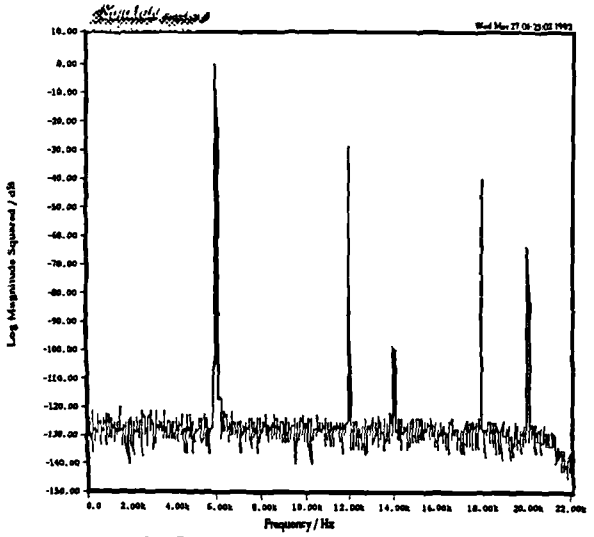


Fig. 7.3b: Baseband Spectrum (UPWM b)

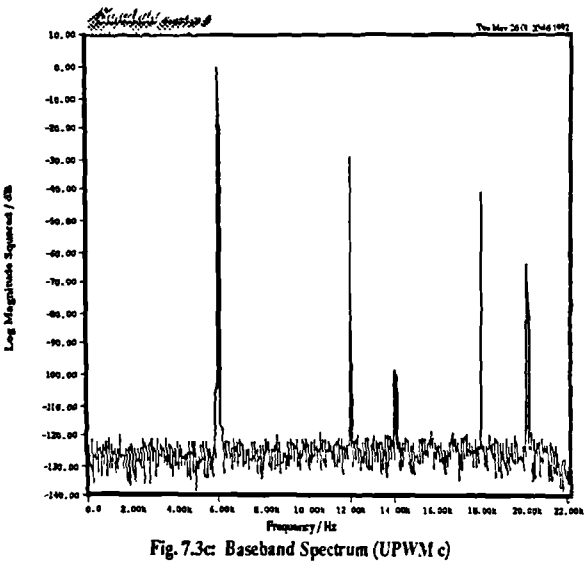


Fig. 7.3c: Baseband Spectrum (UPWM c)

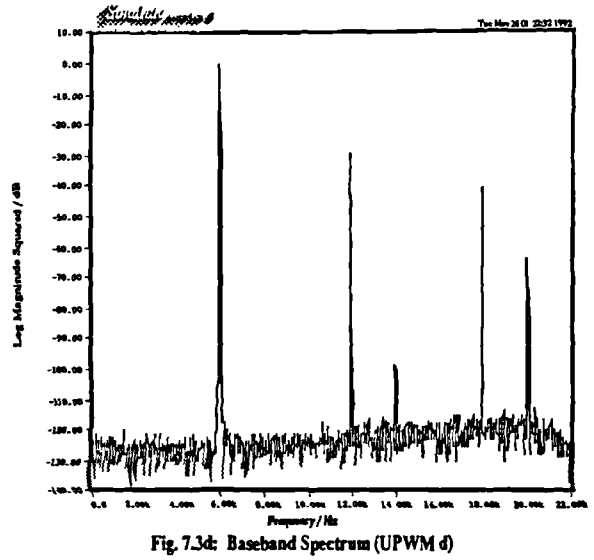


Fig. 7.3d: Baseband Spectrum (UPWM d)

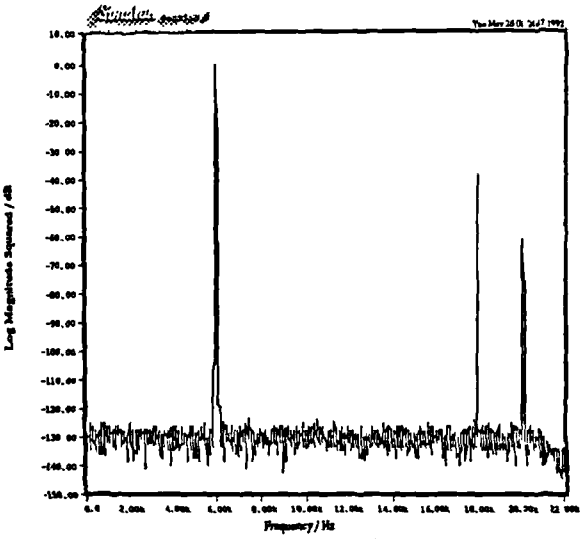


Fig. 7.3e: Baseband Spectrum (UPWM e)

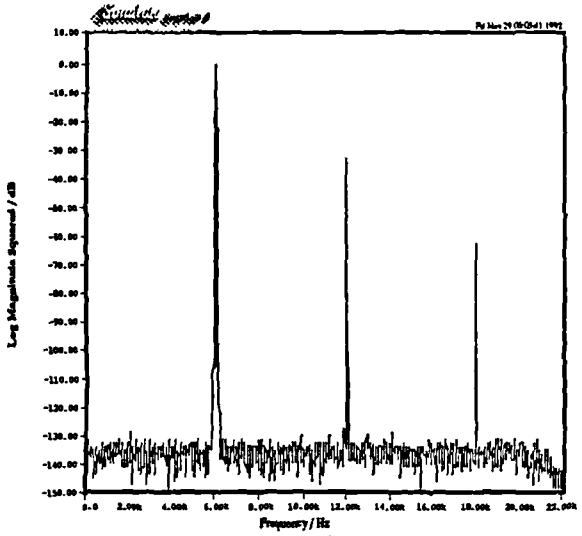


Fig. 7.4a: Baseband Spectrum (w/oversampling UPWM a)

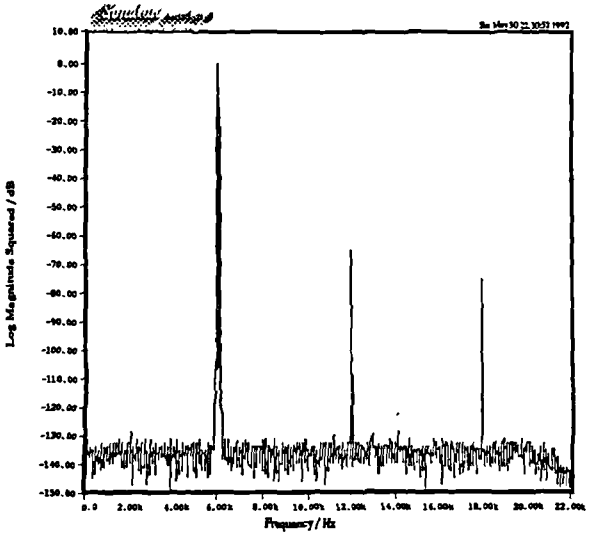


Fig. 7.4b: Baseband Spectrum (w/oversampling UPWM b)

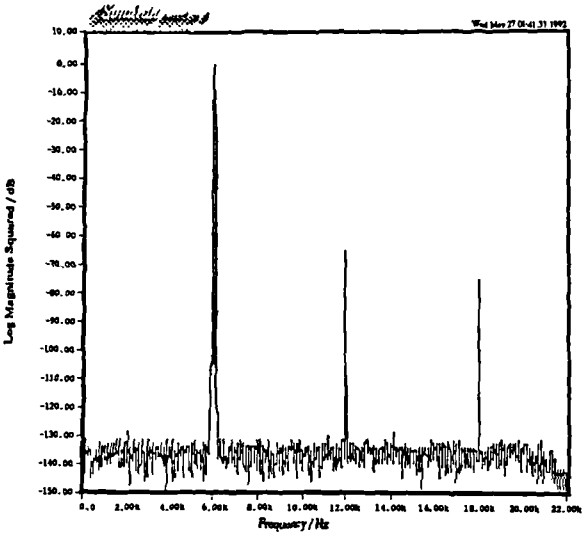


Fig. 7.4c: Baseband Spectrum (w/oversampling UPWM c)

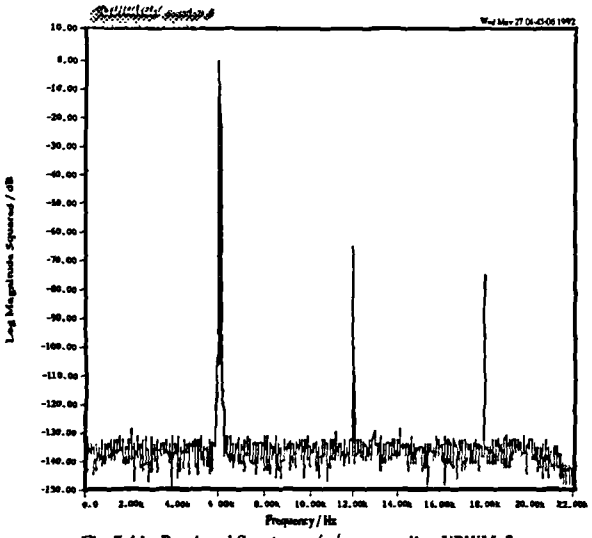


Fig. 7.4d: Baseband Spectrum (w/oversampling UPWM d)

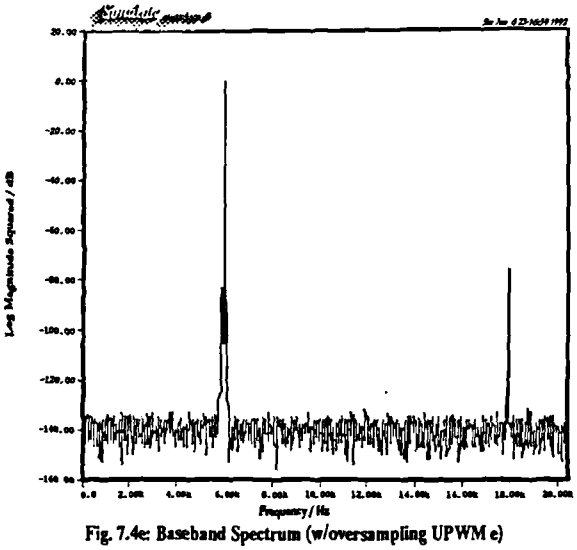


Fig. 7.4e: Baseband Spectrum (w/oversampling UPWM e)

7.2.1 UPWM Harmonic Distortion

The effect of varying the pulse repetition frequency on the levels of harmonic distortion for the five UPWM modulation types is summarized in Table 7.2. Once again the input tone is a 16 bit, 6.001kHz, M=0.8 sinewave. The levels of distortion are all relative to the size of the input tone and are quoted in dB's. We consider relative levels of distortion at the second and third harmonics, which we denote as $\left|\frac{F_2}{F_1}\right|$ and $\left|\frac{F_3}{F_1}\right|$, respectively. From the table we see that for each modulation type the second harmonic is larger than the third harmonic (again except in the case of two sample consecutive modulation, where the second harmonic is zero). We also note that in general the harmonic distortion tends to decrease as the pulse repetition frequency is increased with more dramatic reductions in the third harmonic (about 12dB per doubling of pulse repetition frequency) than in the second (about 6dB per doubling). Also, for a given f_c the harmonic distortion associated with trailing edge modulation is larger than that of the three double sided modulation types while the third harmonic in the two sample consecutive case is at roughly the same level as that of the double sided modulation types.

Varying the modulation depth also has an effect on the relative levels of harmonic distortion. This is shown in Table 7.3 for a 16 bit, 6.001kHz tone input and f_c=44.1kHz. In general, we see that the harmonic distortion decreases as M is decreased. The reductions in the level of the third harmonic are about 12dB per halving of modulation depth and are larger than the roughly 6dB reduction per halving of M for second harmonic. We observe that the absence of the second harmonic in the two sample consecutive case means that, at least for these tests, this modulation type exhibits the lowest total baseband harmonic distortion.

We now verify the correspondence between the results obtained by simulation and those derived either by direct use of the tone spectra equations of Chapter Two or by the corresponding estimates based on approximations to the Bessel function. In Table 7.4 we again consider the case of a 16 bit, 6.001kHz sinewave with peak amplitude 80% of full scale. The pulse repetition frequency is 352.8kHz. We observe from the table that the measured levels are within about half a dB of those predicted by theory. These small errors are caused by the analysis technique (i.e., by the nonidealities of the filters used in the decimation process and by other small effects associated with the DFT). It is also seen that the approximations yield estimates of the relative levels of harmonic distortion which are also within about half a dB of the theoretical levels. Such close approximations are obtained because the high pulse repetition frequency forces the argument of the Bessel

Table 7.2: UPWM Harmonic Distortion as a Function of f_c

($f_0 = 6.001kHz$, $M = 0.800$, $b = 16$)

f_c (L)	Distortion $20\log_{10}(\cdot)$	Modulation Type				
		a	b	c	d	e
44.1*kHz* (1)	$\left\|\frac{F_2}{F_1}\right\|$	-15.37	-28.67	-28.67	-28.67	•
	$\left\|\frac{F_3}{F_1}\right\|$	-27.27	-40.69	-40.69	-40.69	-39.09
88.2*kHz* (2)	$\left\|\frac{F_2}{F_1}\right\|$	-21.70	-41.19	-40.63	-40.63	•
	$\left\|\frac{F_3}{F_1}\right\|$	-39.19	-51.54	-51.33	-51.33	-51.05
176.4*kHz* (4)	$\left\|\frac{F_2}{F_1}\right\|$	-27.27	-53.21	-52.65	-52.65	•
	$\left\|\frac{F_3}{F_1}\right\|$	-50.92	-63.23	-63.04	-63.04	-63.08
352.8*kHz* (8)	$\left\|\frac{F_2}{F_1}\right\|$	-33.22	-64.73	-64.66	-64.66	•
	$\left\|\frac{F_3}{F_1}\right\|$	-62.94	-74.95	-75.06	-75.06	-75.00
705.6*kHz* (16)	$\left\|\frac{F_2}{F_1}\right\|$	-39.80	-76.89	-76.82	-76.82	•
	$\left\|\frac{F_3}{F_1}\right\|$	-75.17	-86.72	-87.01	-87.01	-87.19
1411.2*kHz* (32)	$\left\|\frac{F_2}{F_1}\right\|$	-45.81	-88.84	-88.84	-88.84	•
	$\left\|\frac{F_3}{F_1}\right\|$	-86.95	-96.95	-96.95	-96.95	-96.87

• Distortion does not exist or is beneath 16 bit noise floor

Table 7.3: UPWM Harmonic Distortion as a Function of M

($f_c = 44.1kHz$, $f_v = 6.001kHz$, $b = 16$)

M	Distortion $20\log_{10}(\cdot)$	Modulation Type				
		a	b	c	d	e
0.025	$\left\|\frac{F_2}{F_1}\right\|$	-45.25	-58.82	-58.81	-58.85	•
	$\left\|\frac{F_3}{F_1}\right\|$	-86.85	•	•	•	•
0.050	$\left\|\frac{F_2}{F_1}\right\|$	-39.23	-52.69	-52.68	-52.67	•
	$\left\|\frac{F_3}{F_1}\right\|$	-74.75	-87.66	-87.23	•	86.01
0.100	$\left\|\frac{F_2}{F_1}\right\|$	-33.22	-47.01	-46.68	-46.68	•
	$\left\|\frac{F_3}{F_1}\right\|$	-62.95	-76.89	-76.70	-76.48	-74.93
0.200	$\left\|\frac{F_2}{F_1}\right\|$	-27.72	-40.67	-40.67	-40.67	•
	$\left\|\frac{F_3}{F_1}\right\|$	-50.93	-64.65	-64.66	-64.65	-62.93
0.400	$\left\|\frac{F_2}{F_1}\right\|$	-21.23	-34.66	-34.66	-34.65	•
	$\left\|\frac{F_3}{F_1}\right\|$	-38.97	-52.65	-52.65	-52.66	-50.93
0.800	$\left\|\frac{F_2}{F_1}\right\|$	-15.37	-28.67	-28.67	-28.67	•
	$\left\|\frac{F_3}{F_1}\right\|$	-27.27	-40.69	-40.69	-40.69	-39.09

• Distortion does not exist or is beneath 16 bit noise floor

function to be small compared with its order. As stated in Chapter Two this is necessary for accurate estimates when using the approximations of Table 2.2.

Table 7.4: Correspondence with Theory (UPWM Harmonic Distortion)
($f_c = 352.8kHz$, $f_v = 6.001kHz$, $M = 0.800$, $b = 16$)

Source	Distortion $20\log_{10}(\cdot)$	Modulation Type				
		a	b	c	d	e
Theoretical	$\left\|\frac{F_2}{F_1}\right\|$	-33.41	-64.87	*	*	•
	$\left\|\frac{F_3}{F_1}\right\|$	-63.29	-75.35	*	*	-75.32
Approximation	$\left\|\frac{F_2}{F_1}\right\|$	-33.74	-65.21	*	*	•
	$\left\|\frac{F_3}{F_1}\right\|$	-63.74	-75.78	*	*	-75.76
Simulation	$\left\|\frac{F_2}{F_1}\right\|$	-33.22	-64.73	-64.66	-64.66	•
	$\left\|\frac{F_3}{F_1}\right\|$	-62.94	-74.95	-75.06	-75.06	-75.00

* No theoretical or approximate value available

• Distortion does not exist or is beneath 16 bit noise floor

7.2.2 UPWM Foldback Distortion

Next we turn our attention to the baseband foldback distortion associated with the five modulation types. As discussed in Chapter Two, a particularly severe test for foldback distortion is that of the large, high frequency tone input. For this reason we have chosen a 16 bit, 20.001kHz sinewave input with M=0.8. Table 7.5 shows the effect of varying the pulse repetition frequency on the levels of baseband foldback distortion relative to the input

tone. Recall from Chapter Two that these components may (with some exceptions) arise when $|mf_c + nf_v| < f_b$ where m is a positive integer, n is a negative integer, and f_b is the bandwidth of the system. As with the harmonic distortion, the levels of foldback distortion are relative to the size of the input tone, quoted in dB's, and denoted as $\left|\frac{F_{m+n}}{F_1}\right|$. The spectral components folding back into the baseband from around the carrier frequency (m=1) tend to be the most significant. Those centred about higher harmonics of the carrier generally fall into the baseband at levels well beneath the noise floor. It is seen from the inequality above that varying the pulse repetition frequency will affect the frequencies at which baseband foldback components arise as well as the index, n, associated with these terms. So, for example, in Table 7.5 when f_c=44.1kHz the baseband foldback components centered about the pulse repetition frequency exist at $|1 \cdot 44.1kHz - 2 \cdot 20.001kHz| \approx 4.1kHz$ (for n=–2) and $|1 \cdot 44.1kHz - 3 \cdot 20.001kHz| \approx 15.9kHz$ (for n=–3). In contrast, when f_c=88.2kHz there are components at $|1 \cdot 88.2kHz - 4 \cdot 20.001kHz| \approx 8.2kHz$ (for n=–4) and $|1 \cdot 88.2kHz - 5 \cdot 20.001kHz| \approx 17.8kHz$ (for n=–5). We observe from the table that the foldback distortion associated with trailing edge modulation tends to be the largest where the three double sided modulation types tend to have the smallest amount of total foldback distortion. The two sample consecutive foldback is in between with its (fairly large) single term. Interestingly, we observe that massive reductions in the foldback distortion for all five modulation types are obtained with relatively small increases in pulse repetition frequency. By f_c=176.4kHz the foldback distortion for all five modulation types is beneath the 16 bit noise floor. As explained in Chapter Two, this is due to the *higher order* Bessel functions with small arguments that weight the baseband foldback terms.

Next, we consider the effect of varying the modulation depth associated with the input tone on the levels of baseband foldback distortion. This is shown in Table 7.6 for f_c=44.1kHz, and f_v=20.001kHz. We observe that as with UPWM harmonic distortion, decreasing the modulation depth associated with the input decrease the levels of foldback distortion. There are larger reductions in the ~15.9kHz (n=–3) term (~12dB per halving of M) than in the (–4.1kHz (n=–2) term (~6dB per halving of M).

Lastly, in Table 7.7 for the case of f_c=44.1kHz, f_v=20.001kHz, and M=0.8 we again see close correspondence between theoretical levels of foldback distortion and those obtained by simulation. There are slightly larger differences between the theoretical and approximate levels. This is because for the above system parameters the arguments to the Bessel functions are large compared to their relatively low orders. Hence, the approximations will not be as accurate as in Table 2.4.

Table 7.5: UPWM Baseband Foldback Distortion as a Function of f_c

($f_v = 20.001kHz$, $M = 0.800$, $b = 16$)

f_c (L)	Distortion $20\log_{10}(\cdot)$	Modulation Type				
		a	b	c	d	e
44.1kHz (1)	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-22.80	-44.20	-44.20	-44.20	-29.88
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-27.90	-39.76	-39.76	-39.76	•
88.2kHz (2)	$\left\|\frac{F_{1-4}}{F_1}\right\|$	-83.05	•	•	•	-101.7
	$\left\|\frac{F_{1-5}}{F_1}\right\|$	-102.37	•	•	•	•
176.4kHz (4)	$\left\|\frac{F_{1-8}}{F_1}\right\|$	•	•	•	•	•
	$\left\|\frac{F_{1-9}}{F_1}\right\|$	•	•	•	•	•

• Distortion does not exist or is beneath 16 bit noise floor

7.2.3 UPWM Intermodulation Distortion

We now investigate the performance of the five modulation types with twin tone inputs at frequencies, f_{v_1} and f_{v_2}. In addition to harmonic and foldback distortion terms such input signals also give rise to intermodulation errors at the sum and difference frequencies, $|f_{v_1} \pm f_{v_2}|$, as well as components about the harmonic and foldback distortion terms. We show how the intermodulation products change with pulse repetition frequency. Two tests are used. The first consists of input tones at f_{v_1}=251Hz and f_{v_2}=8.001kHz with M_1=0.8 and M_2=0.2. In the second test f_{v_1}=11.001kHz and f_{v_2}=12.001kHz with M_1=M_2=0.5. Plots of the output spectra for the five modulation types, a through e, for the first test with f_c=44.1kHz are shown in Fig. 7.5a-e. Note the presence of large intermodulation distortion

Table 7.6: UPWM Baseband Foldback Distortion as a Function of M

($f_c = 44.1kHz$, $f_b = 20.001kHz$, $b = 16$)

M	Distortion $20\log_{10}(\cdot)$	Modulation Type				
		a	b	c	d	e
0.025	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-54.29	-74.73	-74.76	-74.24	-60.33
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	•	•	•	•	•
0.050	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-48.33	-68.61	-68.60	-68.48	-54.31
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-77.03	•	•	•	•
0.100	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-42.24	-62.63	-62.63	-62.63	-48.28
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-65.06	-76.23	-76.87	-75.97	•
0.200	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-36.16	-56.58	-56.58	-56.58	-42.25
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-52.93	-64.06	-64.08	-64.06	•
0.400	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-29.88	-50.48	-50.48	-50.48	-36.16
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-40.70	-51.99	-51.98	-51.99	•
0.800	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-22.80	-44.20	-44.20	-44.20	-29.88
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-27.90	-39.76	-39.76	-39.76	•

• Distortion does not exist or is beneath 16 bit noise floor

Table 7.7: Correspondence with Theory (UPWM Foldback Distortion)
($f_c = 44.1kHz$, $f_v = 20.001kHz$, $M = 0.800$, $b = 16$)

Source	Distortion $20\log_{10}(\cdot)$	Modulation Type				
		a	b	c	d	e
Theoretical	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-23.26	-44.67	*	*	-30.34
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-28.31	-40.17	*	*	•
Approximation	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-25.01	-45.36	*	*	-31.04
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-29.78	-40.88	*	*	•
Simulation	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-22.80	-44.20	-44.20	-44.20	-29.88
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-27.90	-39.76	-39.76	-39.76	•

* No theoretical or approximate value available

• Distortion does not exist or is beneath 16 bit noise floor

errors around the 8.001kHz input spaced at multiples of the 251Hz input tone. There is also distortion surrounding the second harmonic of the 8.001kHz input in addition to other terms centred about an ~20.1kHz foldback term. We see that the intermodulation distortion is largest for trailing edge modulation and smallest for two sample consecutive modulation. The three double sided modulation types lie somewhere in between. Results of the same test with f_c=352.8kHz are shown in Fig. 7.6a-e. As could be expected oversampling reduces the intermodulation distortion for all five modulation types.

Next consider the second test with results for f_c=44.1kHz and f_c=352.8kHz given in Fig. 7.7a-e and Fig. 7.8a-e, respectively. Again we see quite severe intermodulation distortion for trailing edge modulation with the three double sided modulation types exhibiting less distortion and two sample consecutive with the lowest distortion. Performance is

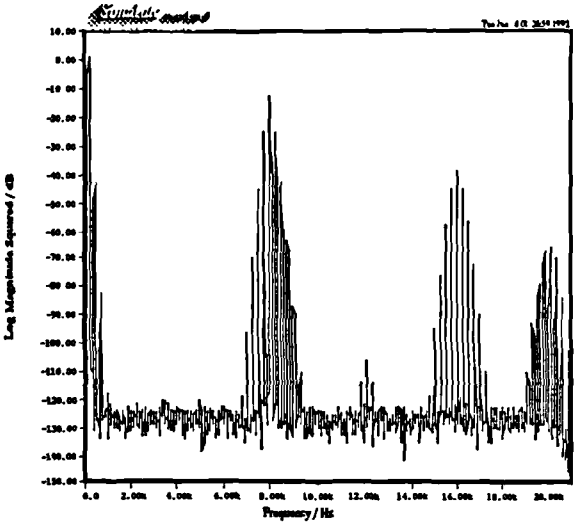


Fig. 7.5a: Intermodulation Distortion (Test 1 UPWM a)

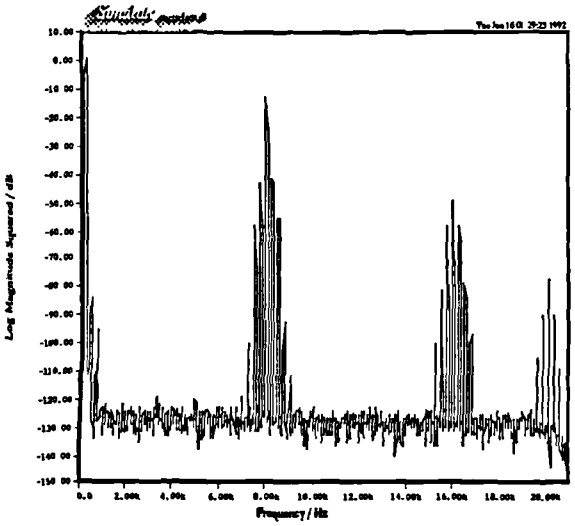


Fig. 7.5b: Intermodulation Distortion (Test 1 UPWM b)

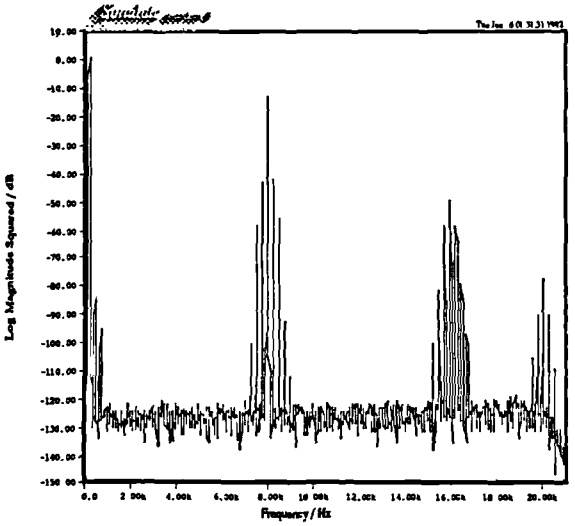


Fig. 7.5c: Intermodulation Distortion (Test 1 UPWM c)

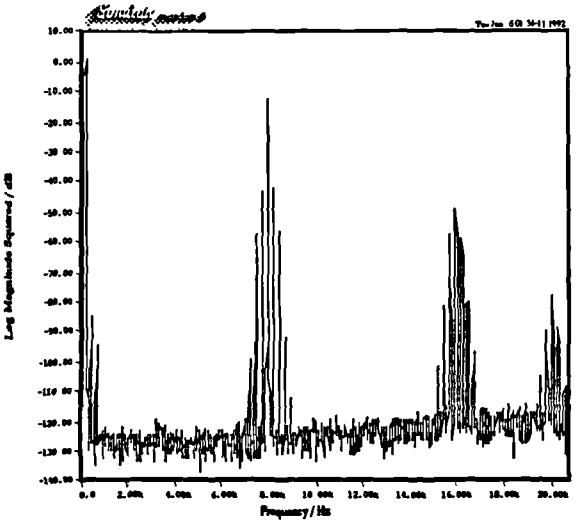


Fig. 7.5d: Intermodulation Distortion (Test 1 UPWM d)

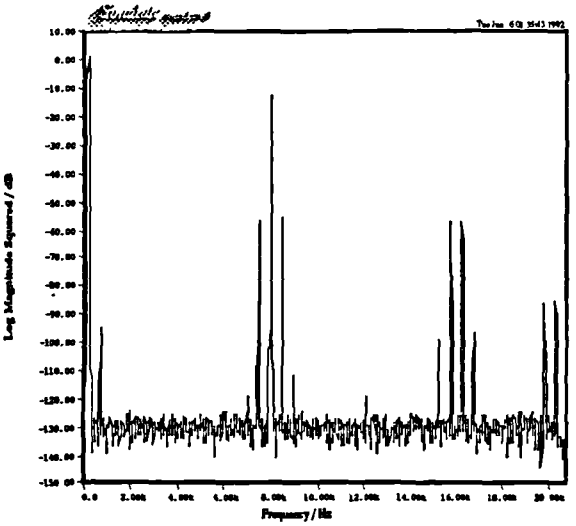


Fig. 7.5e: Intermodulation Distortion (Test 1 UPWM e)

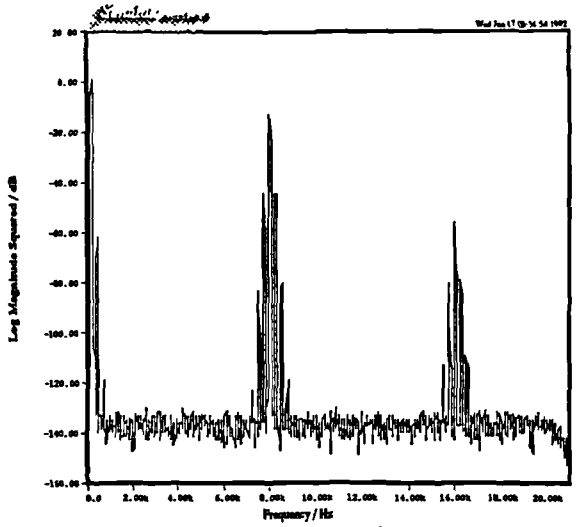


Fig. 7.6a: Intermodulation Distortion (Test 1 w/oversampling UPWM a)

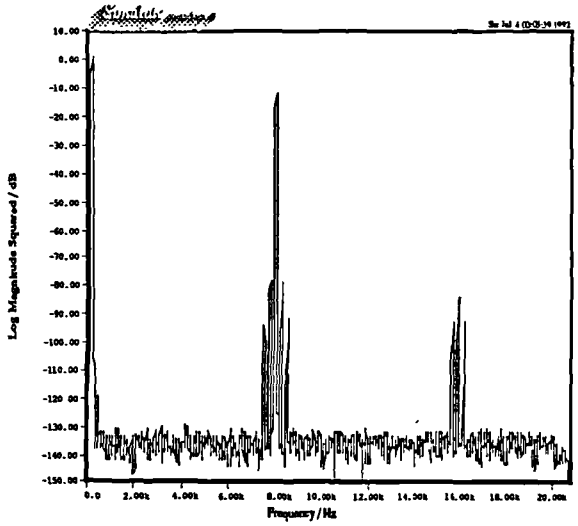


Fig. 7.6b: Intermodulation Distortion (Test 1 w/oversampling UPWM b)

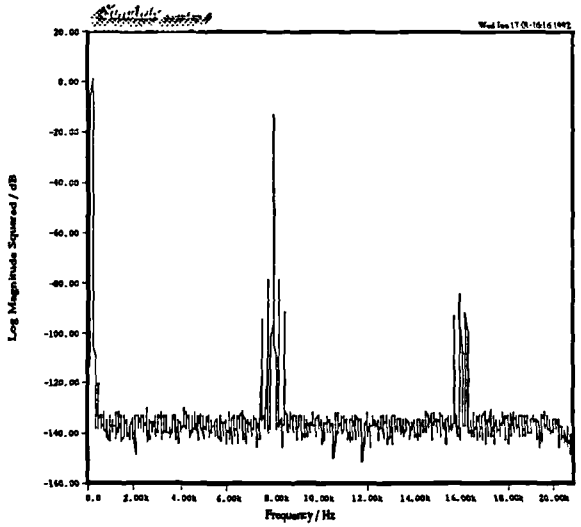


Fig. 7.6c: Intermodulation Distortion (Test 1 w/oversampling UPWM c)

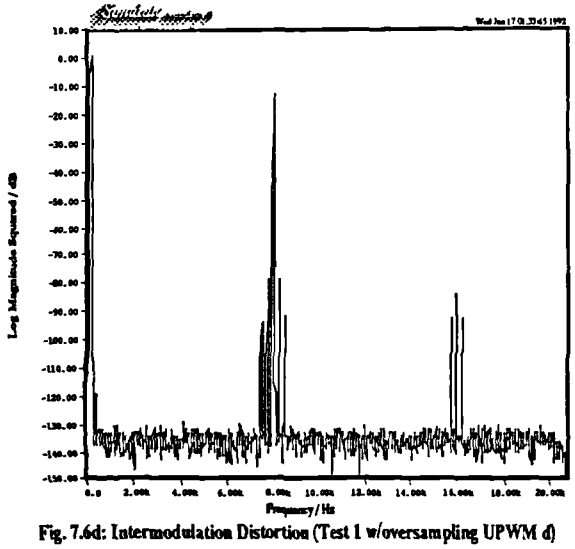


Fig. 7.6d: Intermodulation Distortion (Test 1 w/oversampling UPWM d)

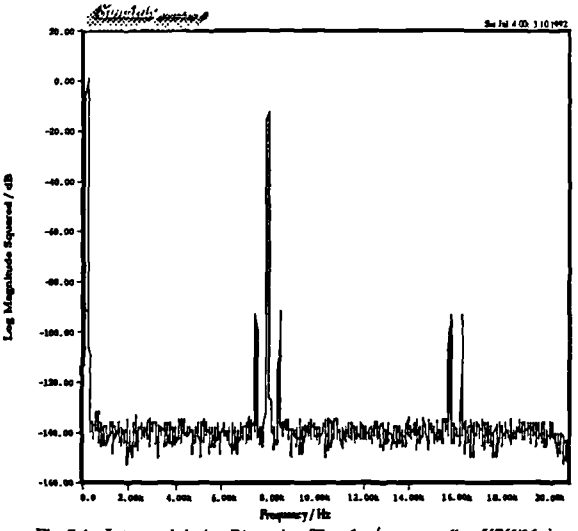


Fig. 7.6e: Intermodulation Distortion (Test 1 w/oversampling UPWM e)

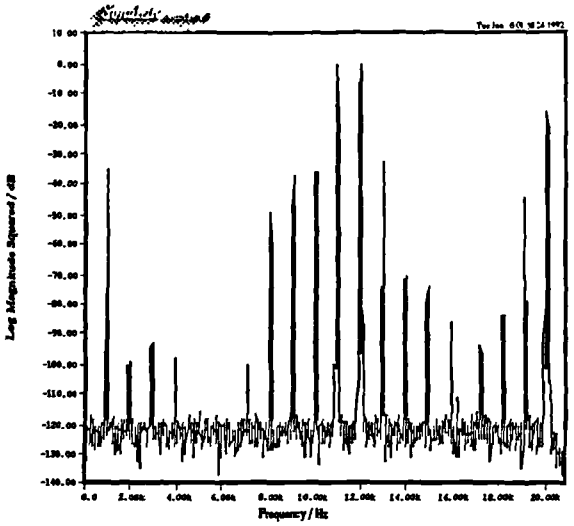


Fig. 7.7a: Intermodulation Distortion (Test 2 UPWM a)

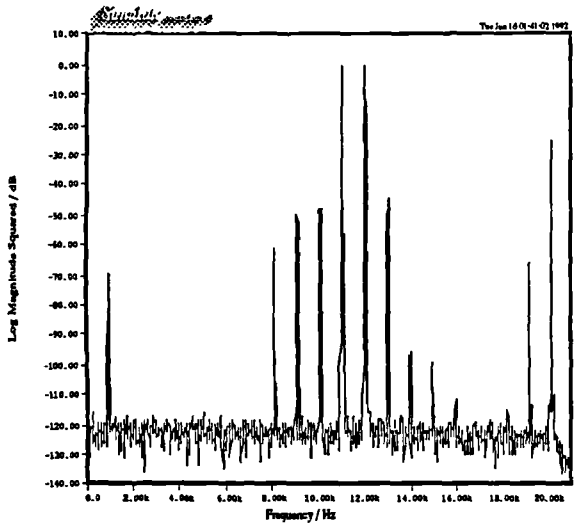


Fig. 7.7b: Intermodulation Distortion (Test 2 UPWM b)

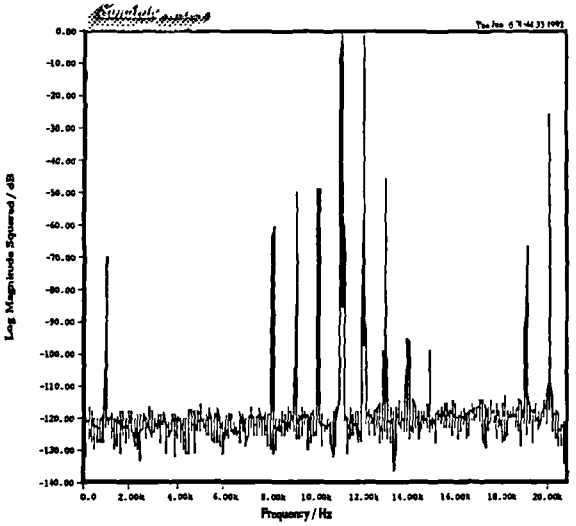


Fig. 7.7c: Intermodulation Distortion (Test 2 UPWM c)

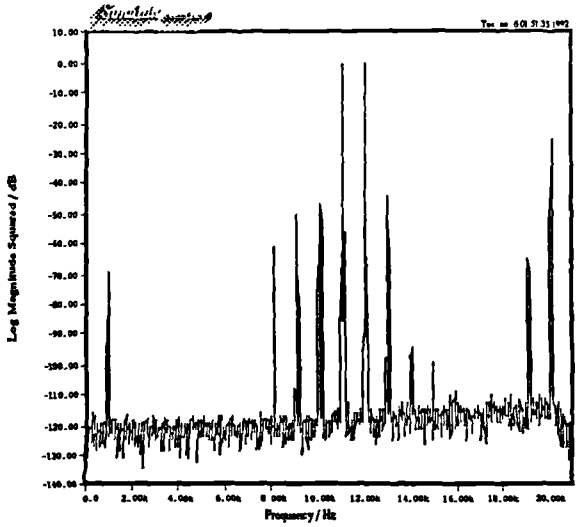


Fig. 7.7d: Intermodulation Distortion (Test 2 UPWM d)

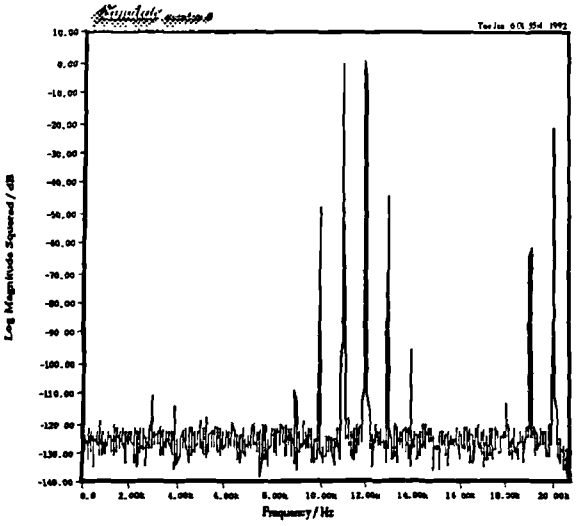


Fig. 7.7e: Intermodulation Distortion (Test 2 UPWM e)

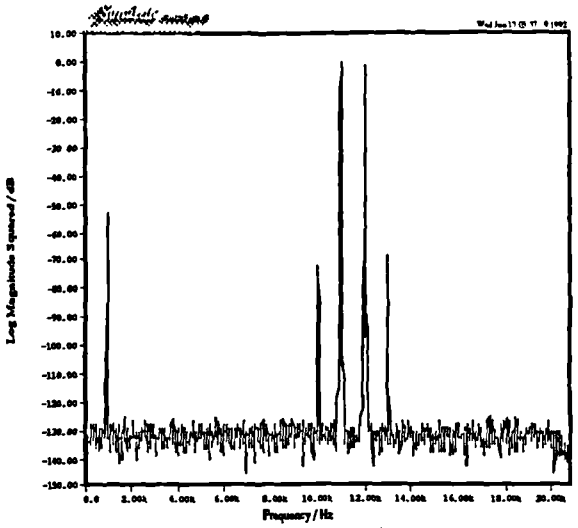


Fig. 7.8a: Intermodulation Distortion (Test 2 w/oversampling UPWM a)

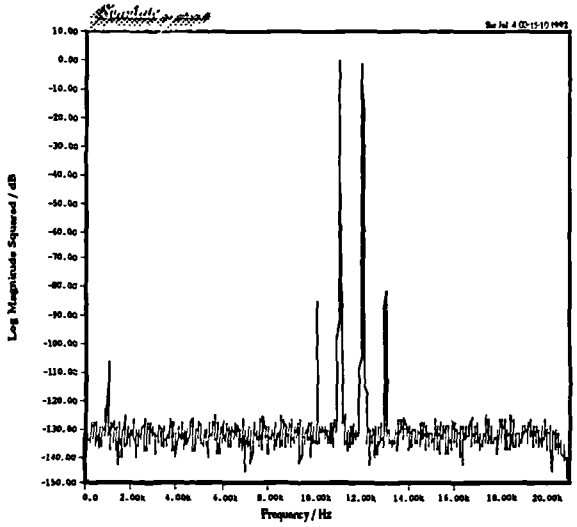


Fig. 7.8b: Intermodulation Distortion (Test 2 w/oversampling UPWM b)

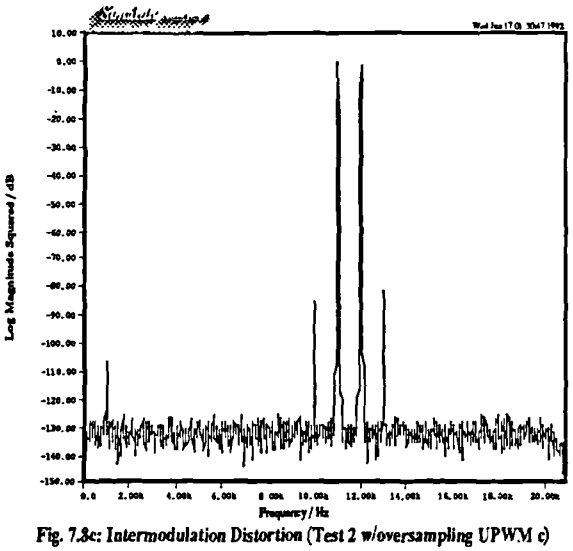


Fig. 7.8c: Intermodulation Distortion (Test 2 w/oversampling UPWM c)

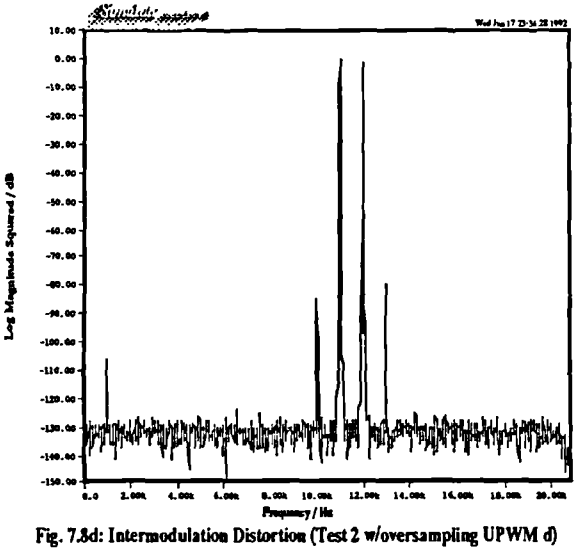


Fig. 7.8d: Intermodulation Distortion (Test 2 w/oversampling UPWM d)

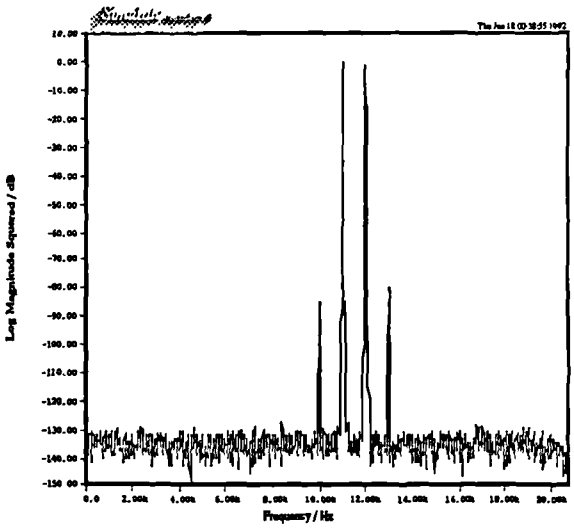


Fig. 7.8e: Intermodulation Distortion (Test 2 UPWM e)

again improved by increasing the pulse repetition frequency.

7.2.4 Overview

Based on the results presented so far we are able to draw a few general conclusions. To begin, it appears that of the five modulation types trailing edge modulation (a) seems to consistently yield the worst results producing levels of distortion much higher than the other four modulation types. The three double sided modulation types (b–d) exhibit higher levels of performance while two sample consecutive UPWM (e) appears to be (with adequate oversampling) the best modulation type—producing the lowest amount of total observed distortion. Also, we have seen that for all five modulation types baseband harmonic, foldback, and intermodulation distortion can each be reduced by increasing the pulse repetition frequency. In addition, we have observed that a reduction in modulation depth results in lower relative levels of baseband harmonic and foldback distortion. Hence performance can also be improved by restricting the maximum modulation depth. However, there are disadvantages associated with both these strategies. As indicated in Chapter Two, practical hardware considerations are such that high quality digital power amplifiers with very large pulse repetition frequencies are difficult to realize. On the other hand, placing severe restrictions on the maximum modulation depth (resulting in very narrow PWM pulses) would drastically limit the output power of the amplifier. Moreover, for a given input wordlength requirement either strategy would lead to higher internal modulator clock speeds. In fact, at this stage, even without these approaches the problem of excessive modulator clock speed presents serious practical difficulties. The next section addresses this issue by presenting results on PWM based DACs which use an oversampled noise shaping network designed to largely overcome this problem.

7.3 ONS/UPWM Based DACs

In this section we present results from computer simulations of ONS/UPWM based DACs as shown in Fig. 7.9. Such systems use a combination of oversampling and noise shaping to simultaneously decrease the modulator clock rate and reduce UPWM distortion. As is seen from the figure, the oversampling is achieved by a two stage interpolator similar to those described in Chapter Three. Also, the noise shaper is of the same form as those in Chapter Four.

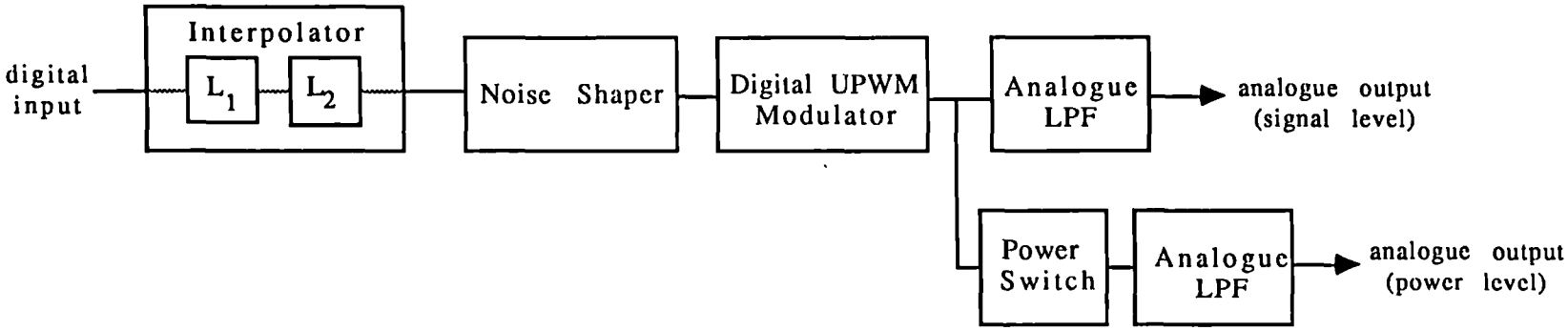


Fig. 7.9: ONS/UPWM DAC

Specifically, we will examine the output tone spectra of ONS/UPWM DACs which use a variety of noise shapers and UPWM modulators. We consider the same six noise shapers introduced in Chapter Four: 12, 10, and eight bit output wordlength systems each with "standard" and minimum phase noise transfer functions (NTFs). (See Appendix 4C.) Recall from Chapter Four with eight times oversampling these noise shapers are all designed to yield approximately 16 bit quality baseband performance. We evaluate the performance of each of the five UPWM modulators when combined with each of the six noise shapers. In all cases the pulse repetition frequency is set to f_c=352.8kHz*, and the input tone is the now familiar 16 bit, f_v=6.001kHz, M=0.80 tone (except when otherwise indicated).

7.3.1 Spectral Plots for "Basic" Systems

We begin with a series of wideband plots shown in Fig. 7.10a-e for the five respective UPWM modulators each driven by the eight bit, fifth order minimum phase NTF noise shaper. (For clarity on such a large frequency range, the input tone frequency was increased to 20kHz.) In all five cases, it is observed that the individual tones present are broadly similar in structure to those of the corresponding non-noise shaped plots presented in the last section. However, there are differing noise floor patterns. Beginning with the trailing edge case, we see that for relatively low frequencies (up to half the pulse repetition frequency) the noise floor seems to mimic the high pass characteristic of the noise transfer function. Similar "images" appear around the carrier, however, this structure is lost for higher multiples of the carrier. For the three double sided modulation types we again have a high pass noise floor up to $\frac{1}{2}f_c$. However, this time the images around the higher harmonics of the carrier can be identified as having a high pass structure but with less pronounced drops in the noise floor than in the low frequency (baseband) region. Interestingly, the less structured high frequency noise floor of the two sample consecutive modulator is more similar to that of the trailing edge modulator than those of the double sided modulators.

We now examine the baseband performance in more detail. Consider first the 12 bit noise shaper with the standard third order NTF. The output spectra of the five 12 bit UPWM modulators (a–e) driven by this noise shaper are shown in Figs. 7.11a-e,

* We note that since the same noise shapers are used for all five modulation types those in the two sample consecutive modulation systems (which operate at twice the rate of the others since for this modulation type f_s=2f_c) are overdesigned (yielding, in effect, a 40kHz baseband).

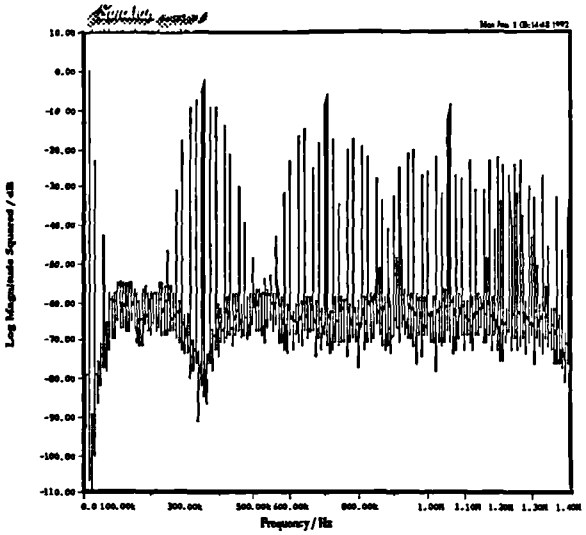


Fig. 7.10a: Wideband Spectrum (ONS MP b'=8 UPWM a)

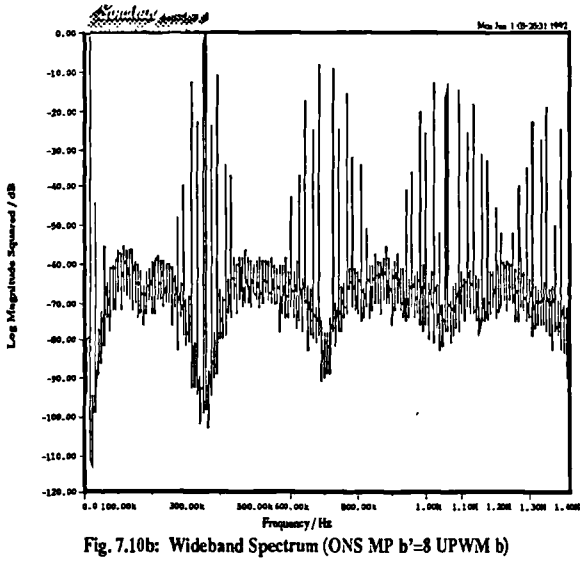


Fig. 7.10b: Wideband Spectrum (ONS MP b'=8 UPWM b)

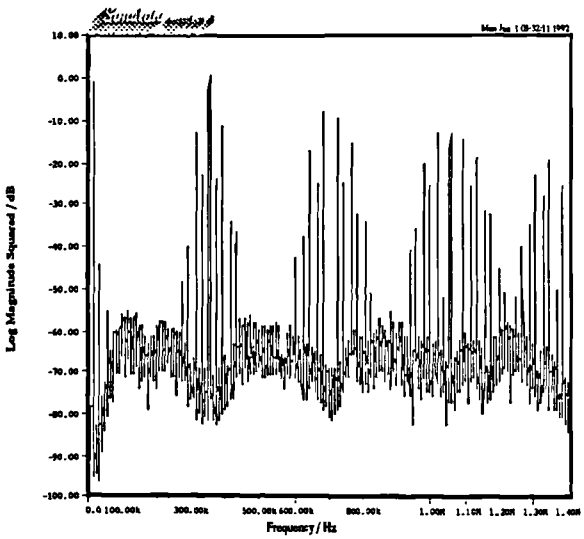


Fig. 7.10c: Wideband Spectrum (ONS MP b'=8 UPWM c)

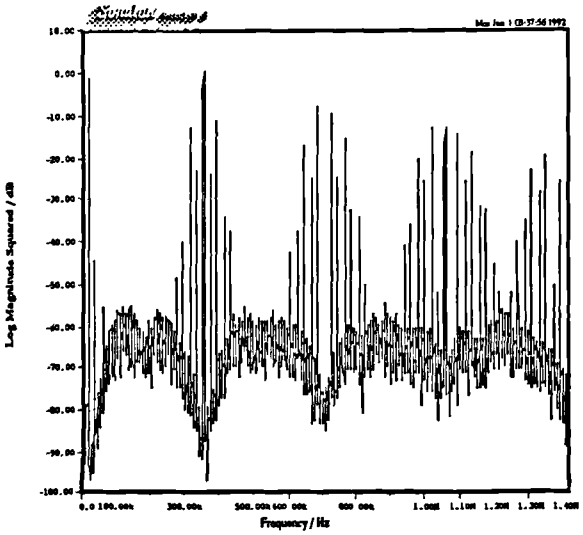


Fig. 7.10d: Wideband Spectrum (ONS MP b'=8 UPWM d)

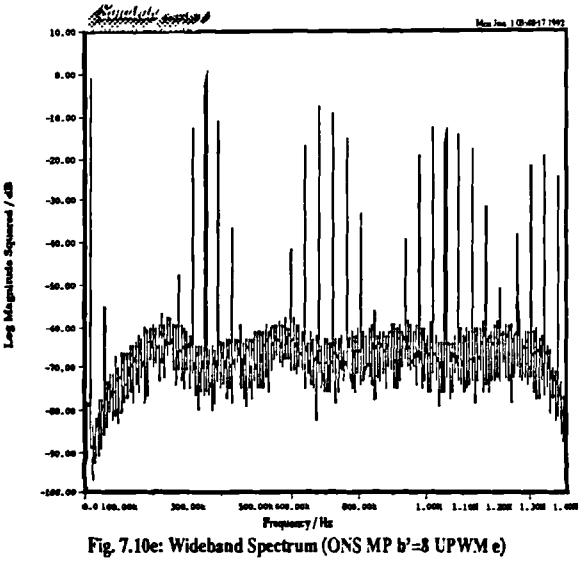


Fig. 7.10e: Wideband Spectrum (ONS MP b'=8 UPWM e)

respectively. They are very similar to the corresponding plots of Fig. 7.4 of the previous subsection which were generated without noise shaping. (Perhaps the most noticeable difference is the lower noise floor in the plots of Fig. 7.4. As discussed earlier this is an effect due to the direct application of data generated at the high sampling rate rather than the use of data interpolated from the Nyquist rate.) When the minimum phase noise transfer function is used the resulting set of plots, shown in Fig. 7.12a-e, is also in close agreement with the non-noise shaped set.

The performance of the 10 bit systems are examined in Figs. 7.13a-e and 7.14a-e for the standard and minimum phase NTF noise shapers, respectively. For the trailing edge (*a*) and double sided symmetric (*b*) cases, the plots are very similar to those of their respective 12 bit cases with little difference between the standard and the minimum phase NTF plots. However, the two fixed double sided asymmetric (*c*) plots both exhibit a distinctive upward sloping noise floor. This is in contrast to the 12 bit case where the three double sided modulation types are very similar looking—all possessing relatively flat noise floors. In Figs. 7.13d and 7.14d we see that the noise penalty encountered in the fixed asymmetric cases is largely eliminated in the alternate asymmetric cases (*d*) with only slight increases in high frequency noise power. In the two sample consecutive modulation (*e*) plots we again note the presence of a high pass noise floor. For the first time we also see the differences created by the choice of NTF. The increase in baseband noise power associated with the standard NTF system is larger than that of the minimum phase NTF system. (The peak level of the noise floor associated with the former is about 6dB higher than that of the latter.)

Figs. 7.15a-e and 7.16a-e show the performance of the five modulation types with eight bit noise shapers. (As mentioned in Chapter Four the modulators in these systems operate at speeds slow enough to be realized in hardware. Hence these simulation results correspond to more realistic ONS/UPWM DACs). Beginning with trailing edge modulation, we again notice a higher frequency baseband noise problem in the system using the standard NTF. However, the use of the minimum phase NTF appears to completely eliminate this effect—yielding a flat noise floor very similar to those of the 12 and 10 bit trailing edge cases. As before, the two double sided symmetric plots possess flat noise floors and generally resemble one another very closely. For fixed asymmetric double sided modulation we again note the presence of increased baseband noise power. While, as in the 10 bit case, there is little difference between the results obtained by using the standard NTF and the minimum phase NTF, we do observe that the peak high frequency noise floor level is significantly (about 12dB) higher in the eight bit plots than in the corresponding 10 bit plots. The alternate asymmetric modulation systems improve the situation somewhat by

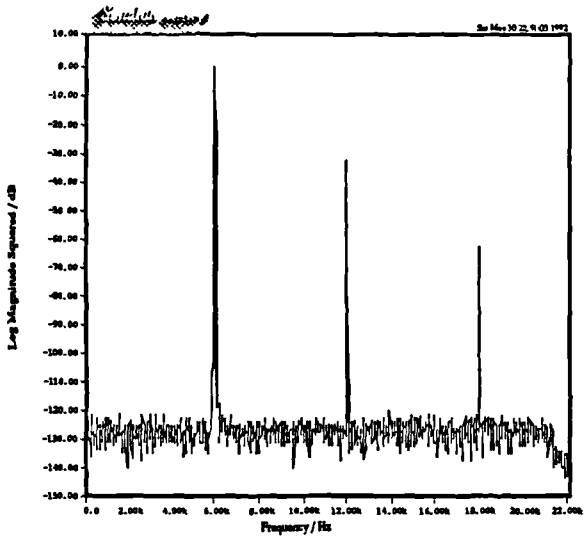


Fig. 7.11a: Baseband Spectrum (ONS Standard b'=12 UPWM a)

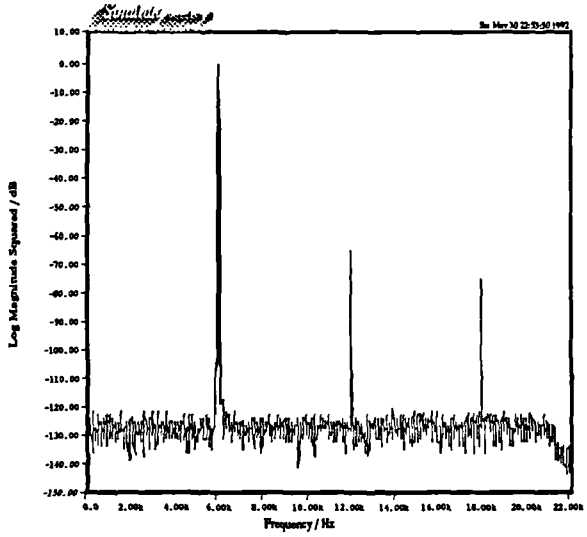


Fig. 7.11b: Baseband Spectrum (ONS Standard b'=12 UPWM b)

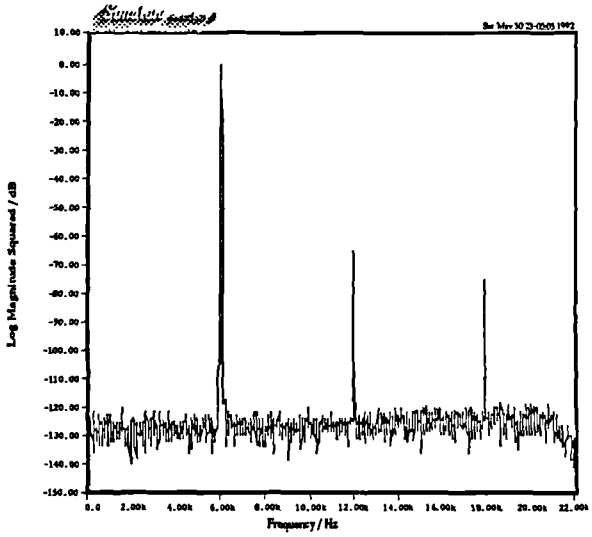


Fig. 7.11c: Baseband Spectrum (ONS Standard b'=12 UPWM c)

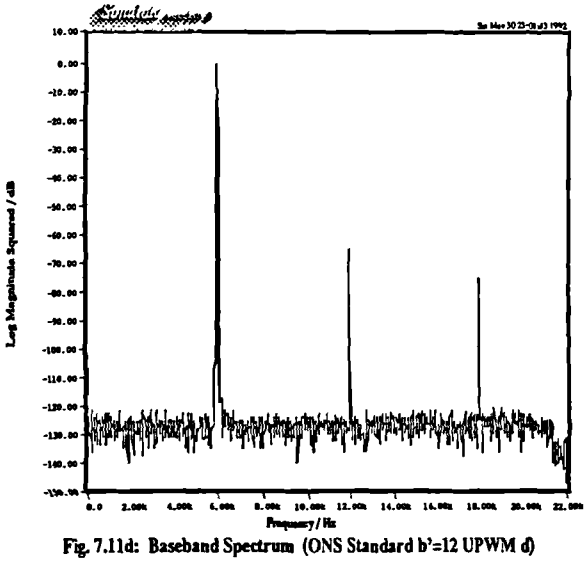


Fig. 7.11d: Baseband Spectrum (ONS Standard b'=12 UPWM d)

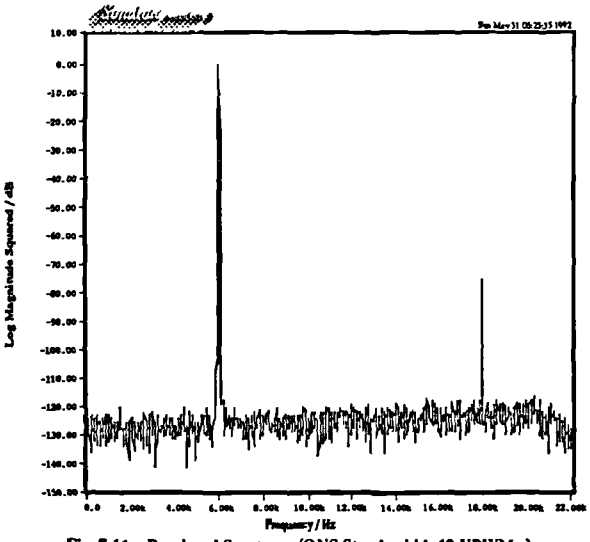


Fig. 7.11e: Baseband Spectrum (ONS Standard b'=12 UPWM e)

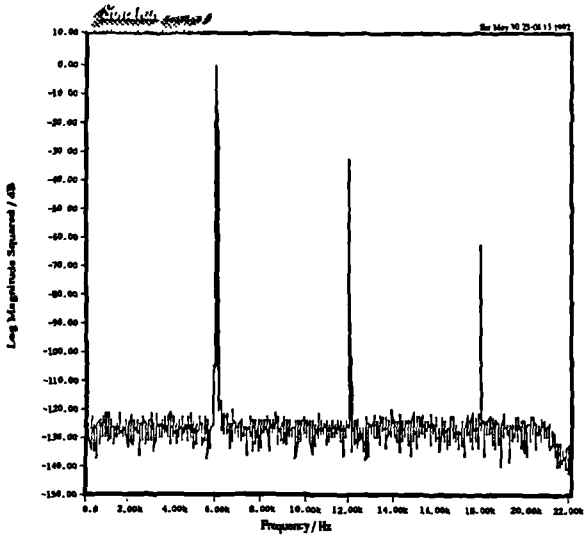


Fig. 7.12a: Baseband Spectrum (ONS MP b'=12 UPWM a)

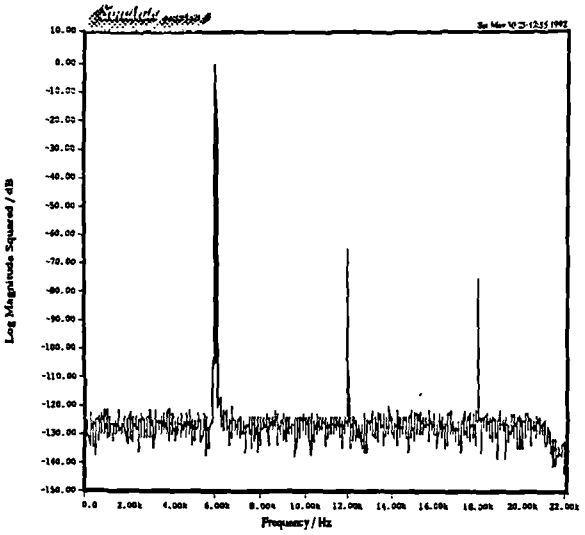


Fig. 7.12b: Baseband Spectrum (ONS MP b'=12 UPWM b)

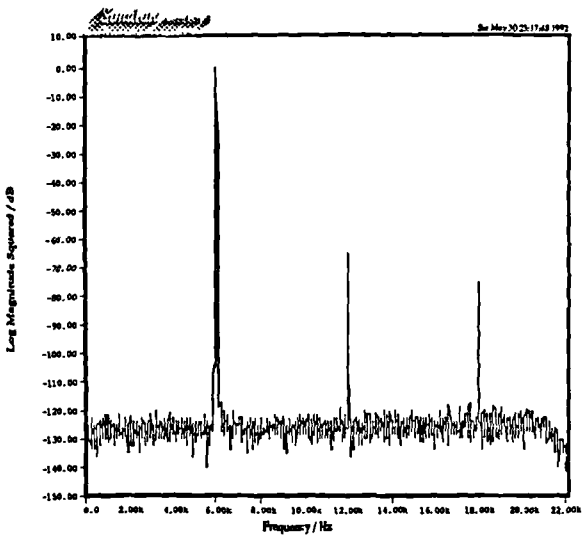


Fig. 7.12c: Baseband Spectrum (ONS MP b'=12 UPWM c)

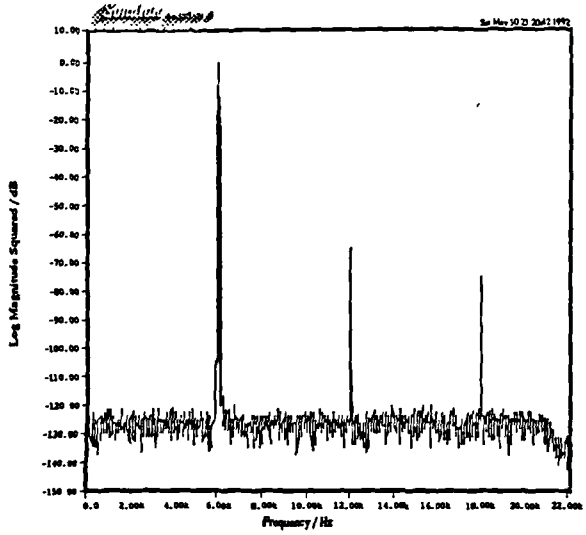


Fig. 7.12d: Baseband Spectrum (ONS MP b'=12 UPWM d)

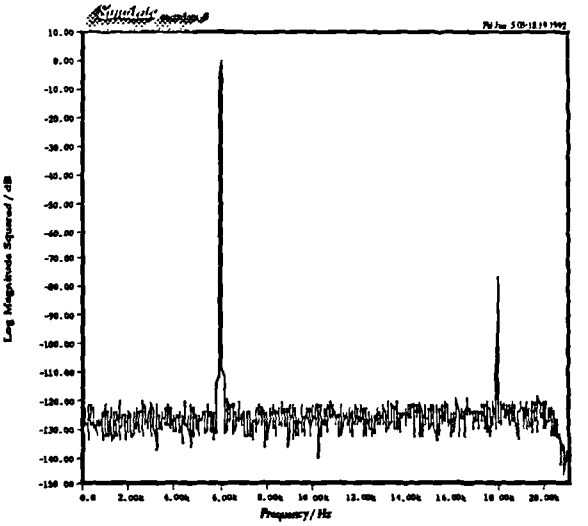


Fig. 7.12e: Baseband Spectrum (ONS MP b'=12 UPWM e)

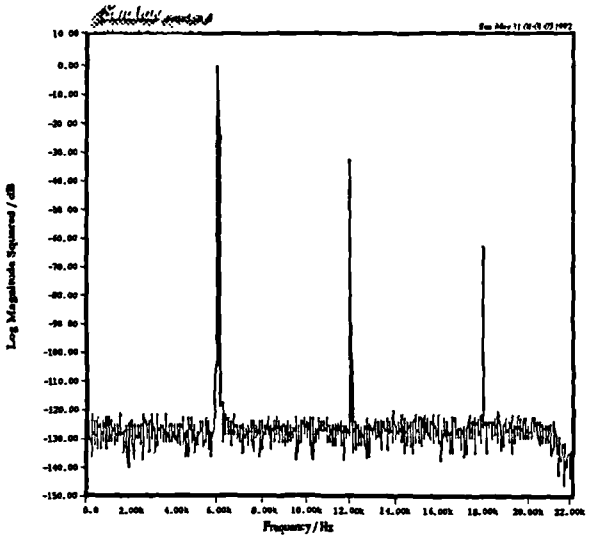


Fig. 7.13a: Baseband Spectrum (ONS Standard b'=10 UPWM a)

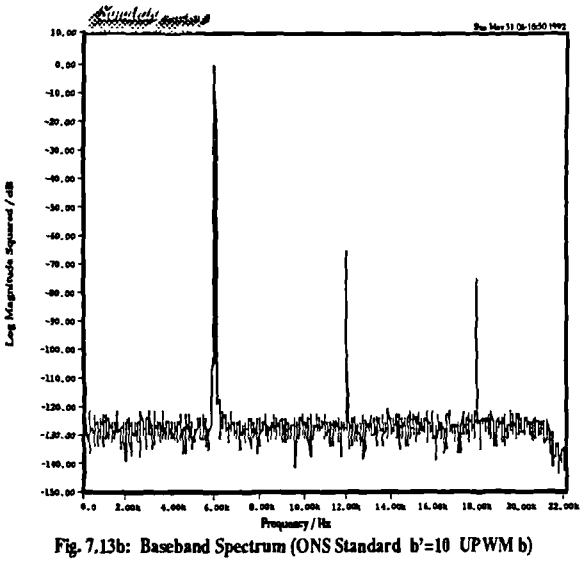


Fig. 7.13b: Baseband Spectrum (ONS Standard b'=10 UPWM b)

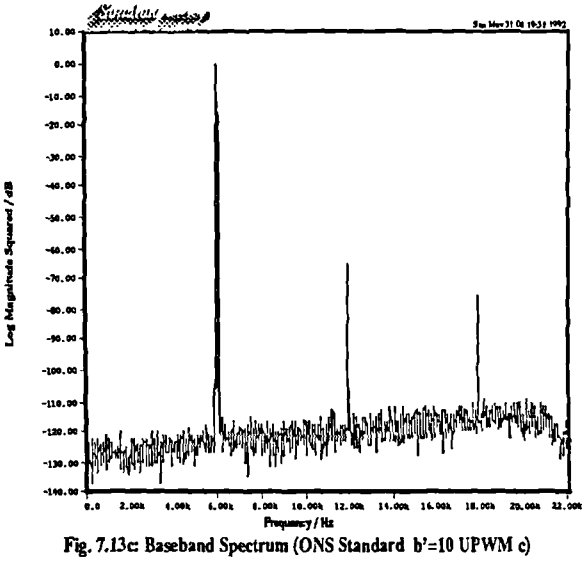


Fig. 7.13c: Baseband Spectrum (ONS Standard b'=10 UPWM c)

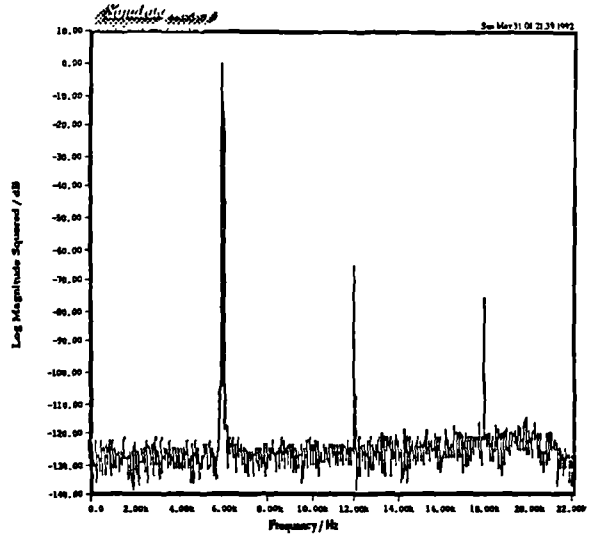


Fig. 7.13d: Baseband Spectrum (ONS Standard b'=10 UPWM d)

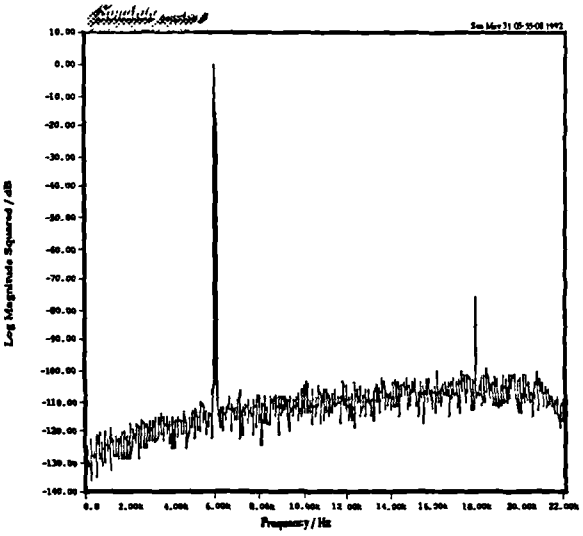


Fig. 7.13e: Baseband Spectrum (ONS Standard b'=10 UPWM e)

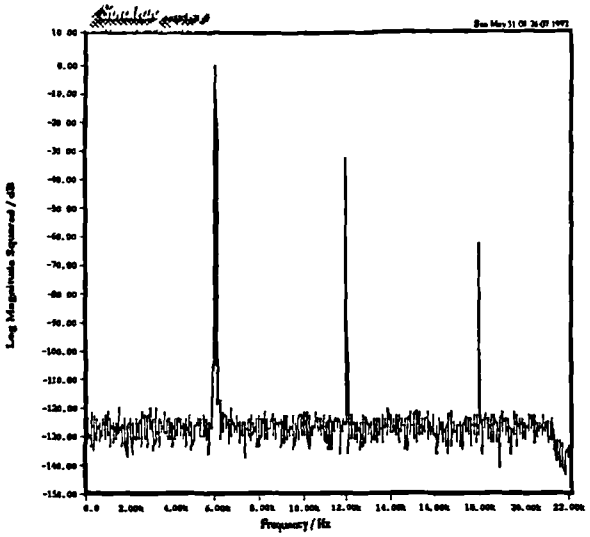


Fig. 7.14a: Baseband Spectrum (ONS MP b'=10 UPWM a)

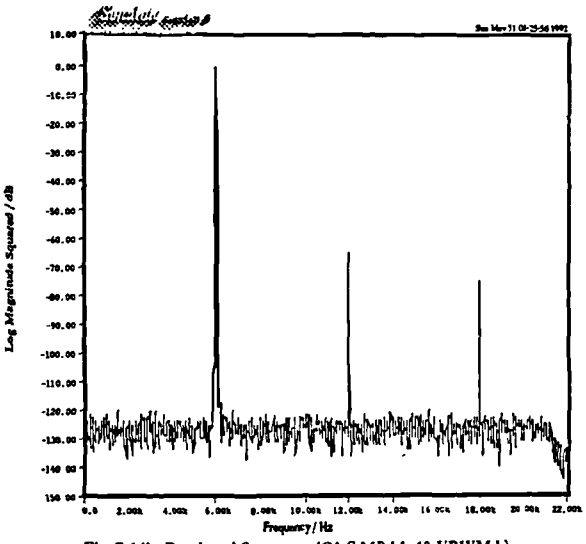


Fig. 7.14b: Baseband Spectrum (ONS MP b'=10 UPWM b)

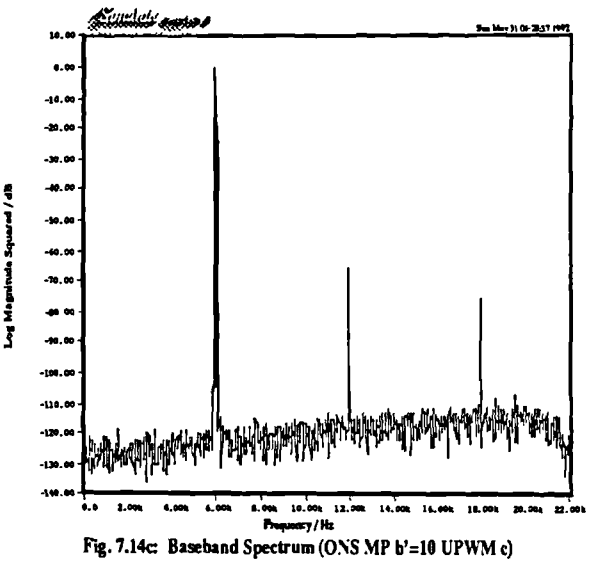


Fig. 7.14c: Baseband Spectrum (ONS MP b'=10 UPWM c)

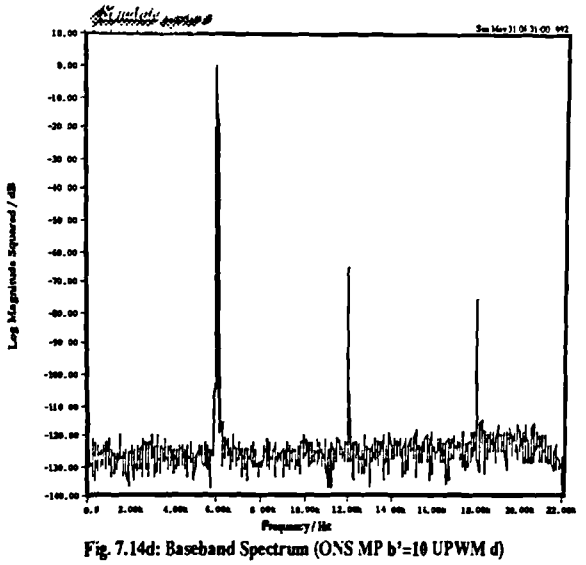


Fig. 7.14d: Baseband Spectrum (ONS MP b'=10 UPWM d)

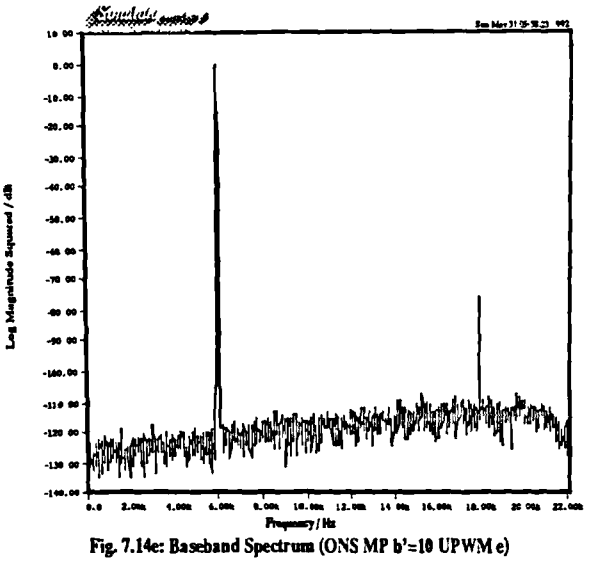


Fig. 7.14e: Baseband Spectrum (ONS MP b'=10 UPWM e)

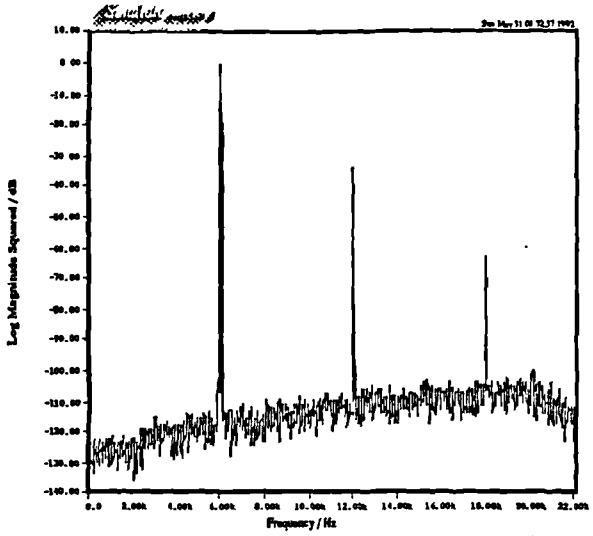


Fig. 7.15a: Baseband Spectrum (ONS Standard b'=8 UPWM a)

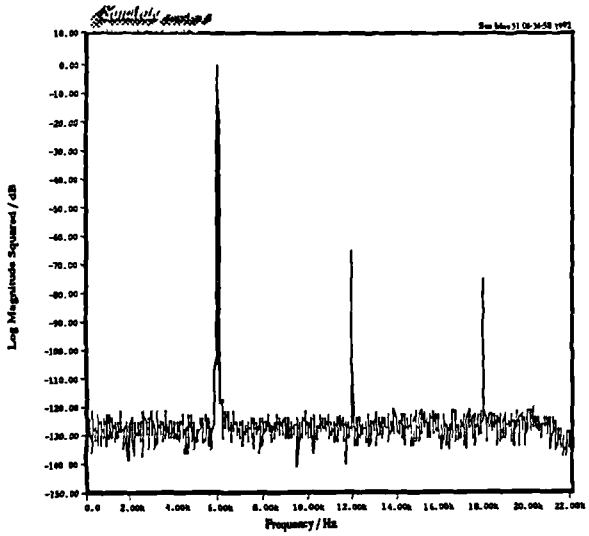


Fig. 7.15b: Baseband Spectrum (ONS Standard b'=8 UPWM b)

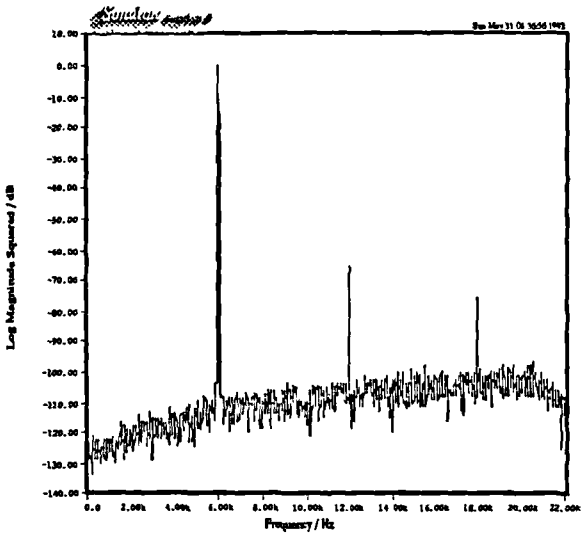


Fig. 7.15c: Baseband Spectrum (ONS Standard b'=8 UPWM c)

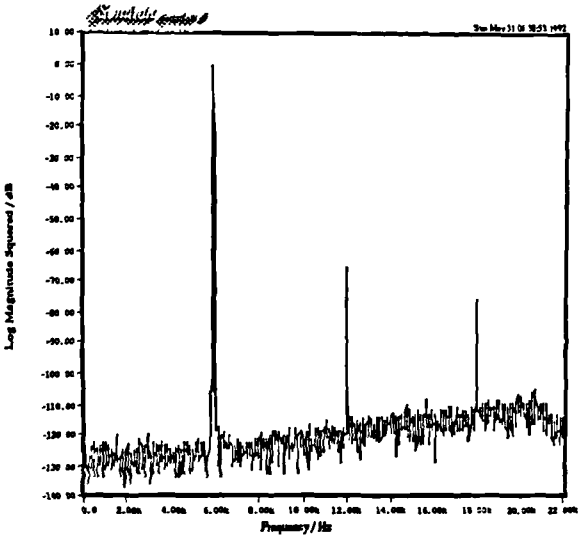


Fig. 7.15d: Baseband Spectrum (ONS Standard b'=8 UPWM d)

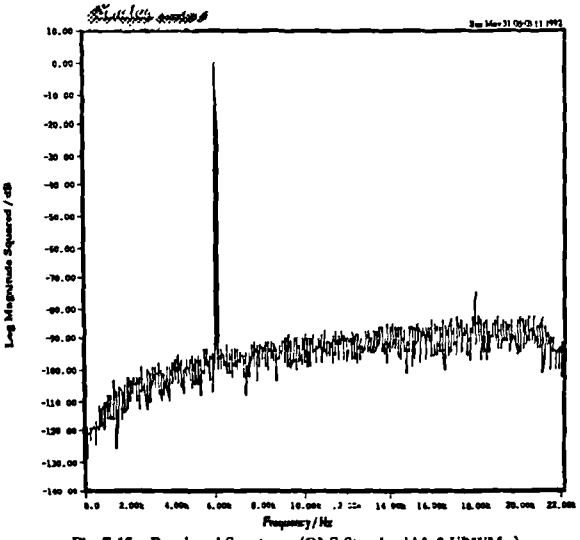


Fig. 7.15e: Baseband Spectrum (ONS Standard b'=8 UPWM e)

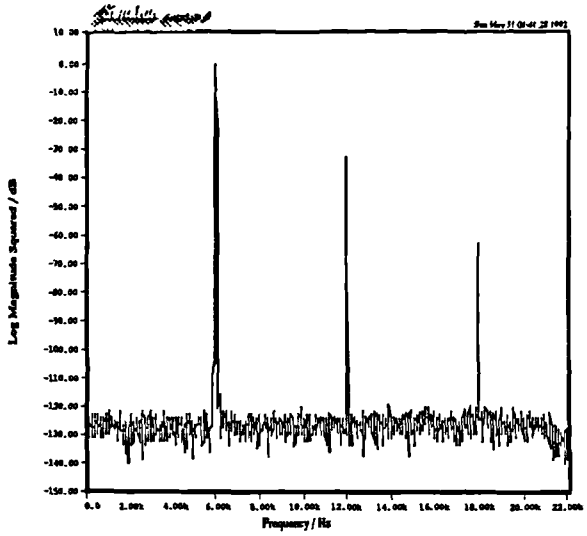


Fig. 7.16a: Baseband Spectrum (ONS MP b'=8 UPWM a)

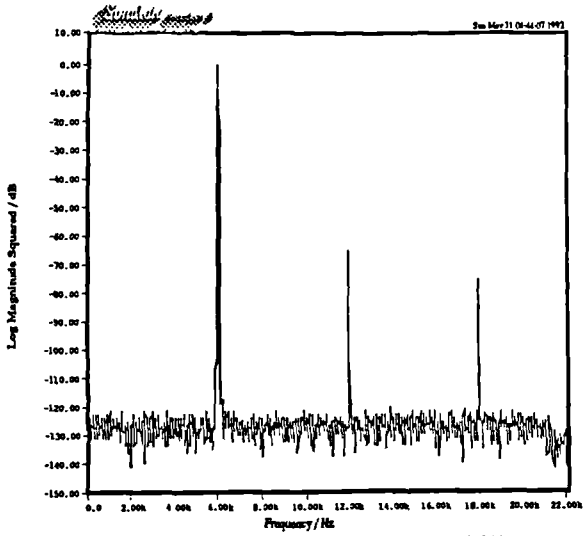


Fig. 7.16b: Baseband Spectrum (ONS MP b'=8 UPWM b)

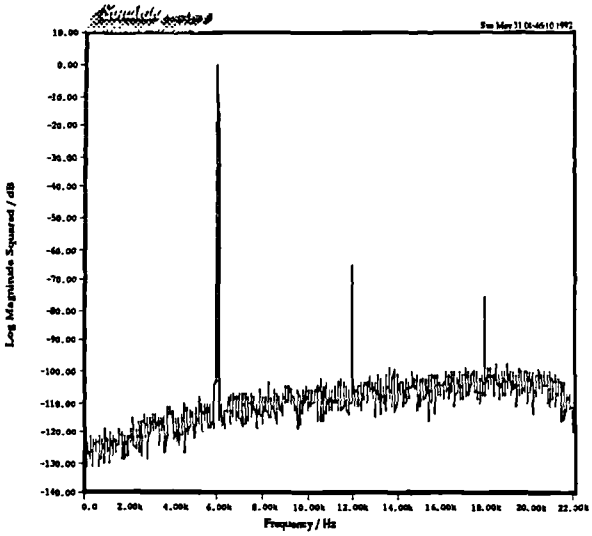


Fig. 7.16c: Baseband Spectrum (ONS MP b'=8 UPWM c)

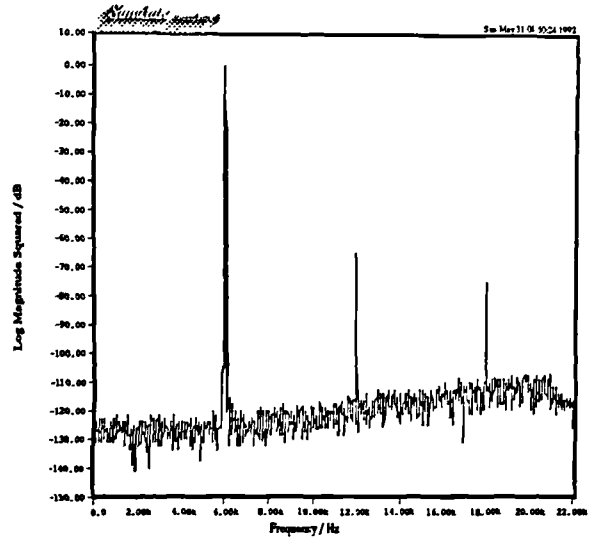


Fig. 7.16d: Baseband Spectrum (ONS MP b'=8 UPWM d)

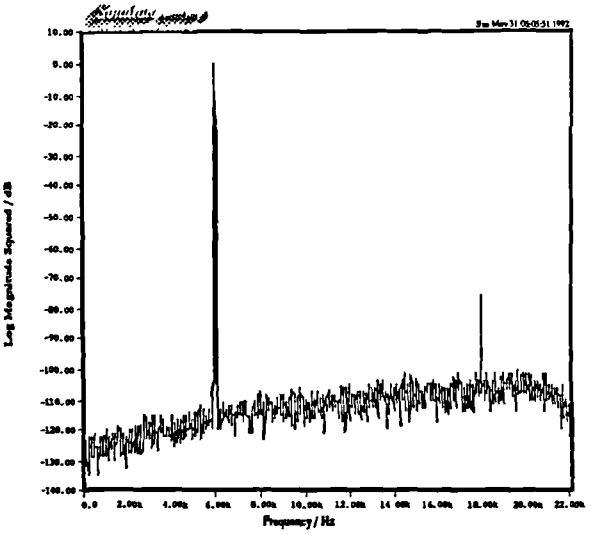


Fig. 7.16e: Baseband Spectrum (ONS MP b'=8 e)

lowering the peak noise floor level of both the standard and minimum phase NTF systems to levels somewhat beneath (about 6dB beneath) those in the fixed asymmetric cases, however, the high pass shape of the noise floor is still easily observed. For the two sample consecutive plots we again notice a high frequency noise penalty which is particularly severe in the standard NTF case.

7.3.2 Explanations and Solutions for the Increased Noise Power Problem

What is responsible for the strange effects we observed in the eight and 10 bit plots? For systems with such complicated nonlinearities it is extremely difficult to formulate precise quantitative explanations. Instead we try to offer a few intuitive, qualitative comments with the proviso that further work is necessary.

Let us first consider the increased baseband noise power associated with some of the trailing edge spectra. As alluded to in Chapter Four, this effect is believed to be caused by the modulation process redistributing some of the very high frequency (well above baseband) noise power back into the baseband. It is tempting to draw an analogy between this effect and that of PWM foldback or intermodulation distortion for tone inputs. We can think of this excess baseband noise as being a result of the higher frequency (above baseband) noise shaper output noise folding back and/or intermodulating with itself back into the baseband. Earlier in this chapter we saw that high amplitude, high frequency tones were capable of producing large baseband foldback and intermodulation distortion terms. Likewise we saw that in the eight bit case the standard NTF system, which produces *much more* high frequency noise power than the minimum phase NTF system, was also responsible for some of the observed increased noise power effects. Although further research is necessary, this analogy also offers an explanation as to why the effect is less severe in the 10 and 12 bit cases where there is generally less high frequency noise power than in the eight bit standard NTF case. Also, in the same way we saw that single sided foldback and intermodulation distortion is much larger than that of double sided modulation, comparison of the trailing edge and double sided symmetric plots shows that the effect is much more pronounced in the former. Lastly, as baseband foldback and intermodulation distortion for tone inputs can be made less severe with additional oversampling we may expect the same to be true of this effect. This is shown to be the case in Fig. 7.17a where we have used a fifth order standard NTF with eight bit noise shaping and a pulse repetition frequency of 16 times the Nyquist rate (f_s=705.6kHz). Comparing this figure with the same system at

eight times oversampling (Fig. 7.15a) shows a substantial reduction in the increase of baseband noise power.

As in the eight times oversampled case, Fig. 7.18a shows that the fifth order optimized NTF system operating at 16 times oversampling does not exhibit any noticeable increase in noise power. The benefits obtained by oversampling in the standard NTF case are believed to be due to the the increased bandwidth over which these unwanted effects can be spread. Interestingly, we note that while for tone modulation doubling the pulse repetition frequency often resulted in large reductions in baseband foldback distortion, reductions of this size are not evident under the current effect. This is perhaps because for tone modulation oversampling places a greater gap in frequency between the baseband and the carrier. As described in Chapter Two, it is this gap which causes the reduction in baseband foldback distortion. However, in this case, with the high frequency noise shaper error power always extending to half the carrier rate, such gaps never arise.

Next we look at the three double sided modulation types. At present, the effects encountered in these systems are not well understood and are the subject of continuing research. With this qualification, we begin by noting the complete absence of increased noise power in the symmetric modulation plots in all cases. Clearly then, the increased baseband noise power in the asymmetric modulation types is due to the asymmetries in the PWM pulse rather than any problem inherent to double sided modulation. Now recall from Chapter Two that for even valued input samples both forms of asymmetric modulation (c and d) produce pulses which are identical in width and position to those produced by symmetric modulation. However, for odd valued input samples, while the associated pulse widths are the same as those produced by symmetric modulation, the pulse positions are shifted in time. (See Fig. 2.5 of Chapter Two.) The phase errors introduced by these intermittent shifts manifest themselves as increased noise power at the output of the system. When considering the errors associated with individual pulses in isolation, it does not matter if the position of the pulse is delayed or is advanced. However, over many pulses the cumulative effects appear such that always advancing (or delaying) the pulse for odd input values results in higher baseband output noise power than alternately advancing or delaying the asymmetric pulses. The later seems to result in a partial cancellation of the effect. The errors introduced by advancing (delaying) a pulse are partly compensated for by delaying (advancing) the next pulse associated with an odd valued input sample.

While this explanation has its appeal we have actually found the opposite to be true for systems which do not use noise shaping (i.e., fixed asymmetric modulation yields better results than alternate asymmetric modulation when there is no noise shaper). In fact, careful inspection of Figs. 7.3c-d of the previous section indicates a slight relative increase

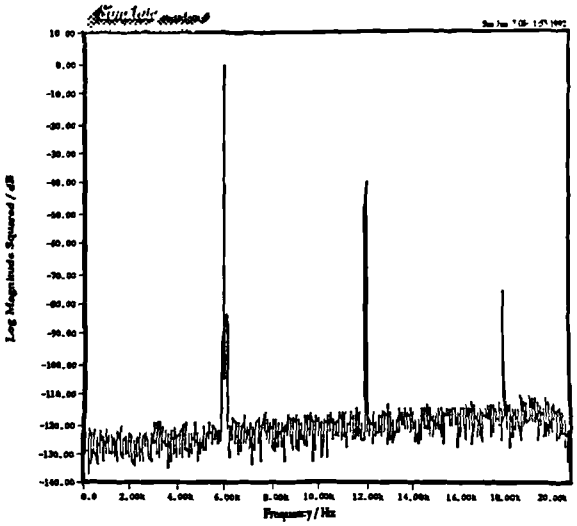


Fig. 7.17a: Baseband Spectrum (16X ONS Standard b'=12 UPWM a)

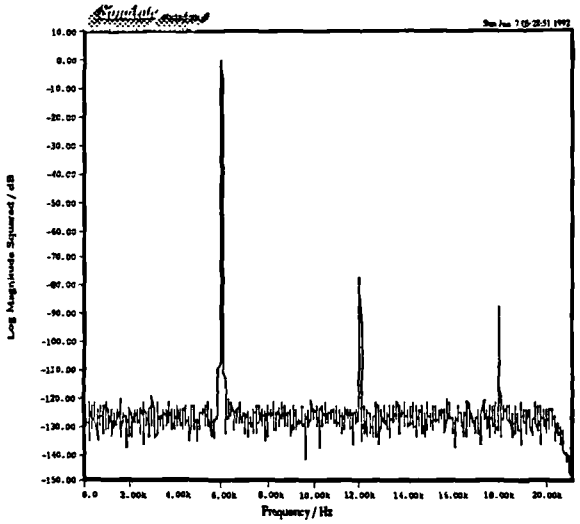


Fig. 7.17b: Baseband Spectrum (16X ONS Standard b'=8 UPWM b)

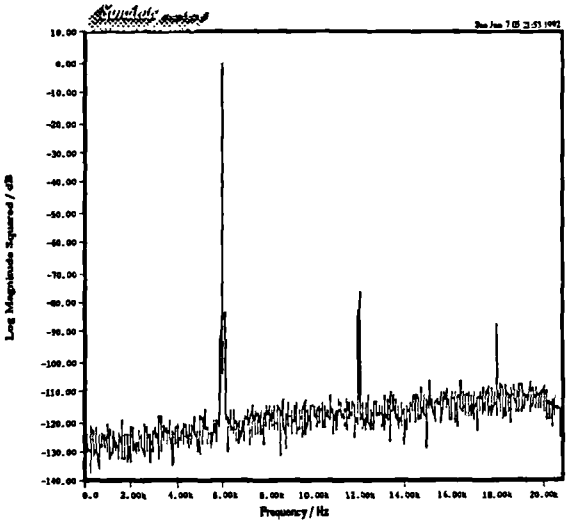


Fig. 7.17c: Baseband Spectrum (16X ONS Standard b'=8 UPWM c)

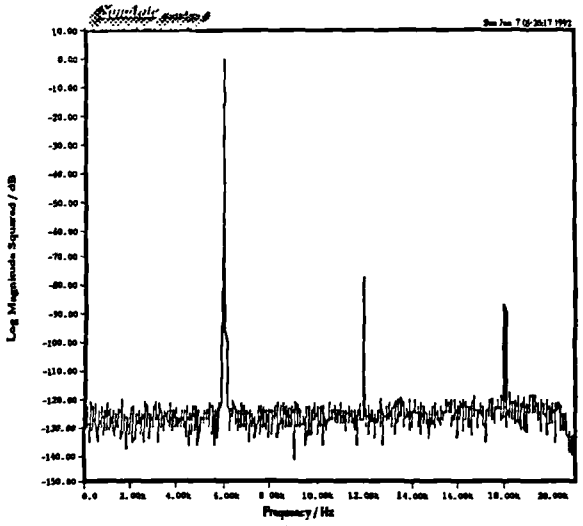


Fig. 7.17d: Baseband Spectrum (16X ONS Standard b'=8 UPWM d)

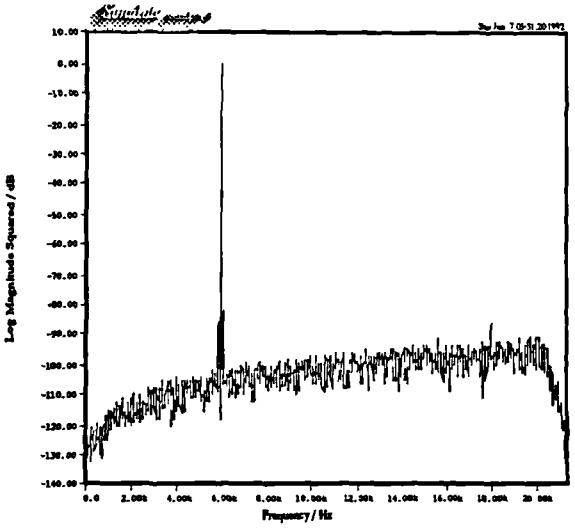


Fig. 7.17e: Baseband Spectrum (16X ONS Standard b'=8 UPWM e)

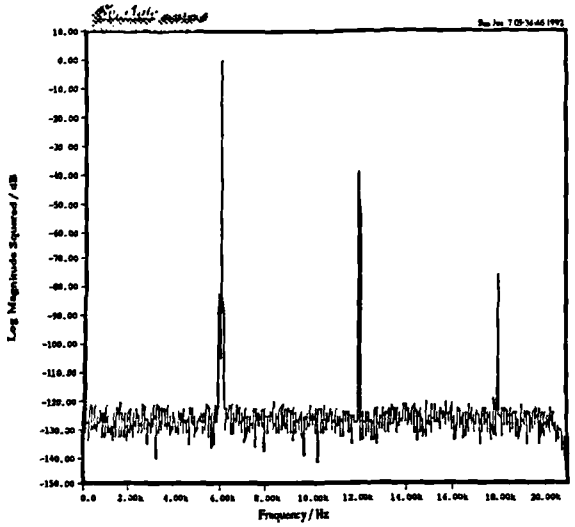


Fig. 7.18a: Baseband Spectrum (16X ONS MP b'=8 UPWM a)

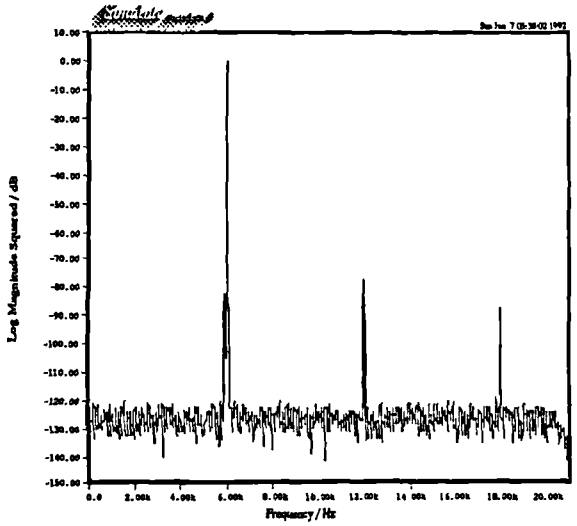


Fig. 7.18b: Baseband Spectrum (16X ONS MP b'=8 UPWM b)

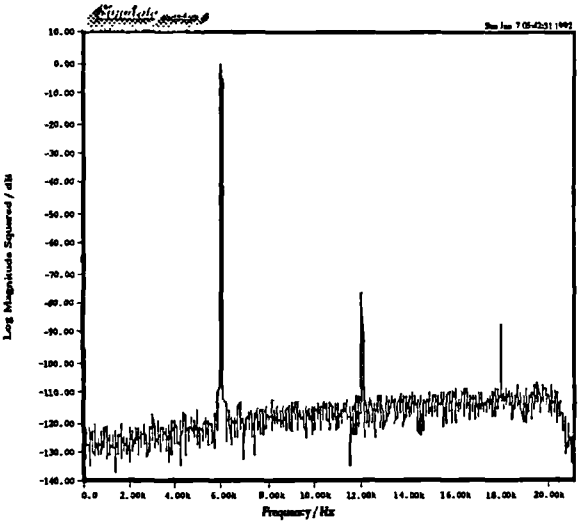


Fig. 7.18c: Baseband Spectrum (16X ONS MP b'=8 UPWM c)

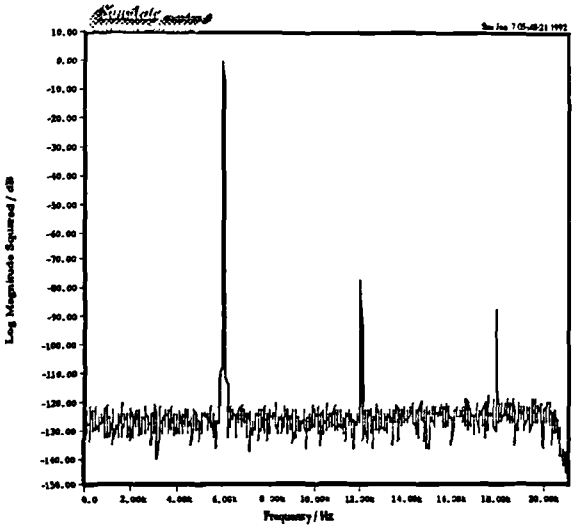


Fig. 7.18d: Baseband Spectrum (16X ONS Mp b'=8 UPWM d)

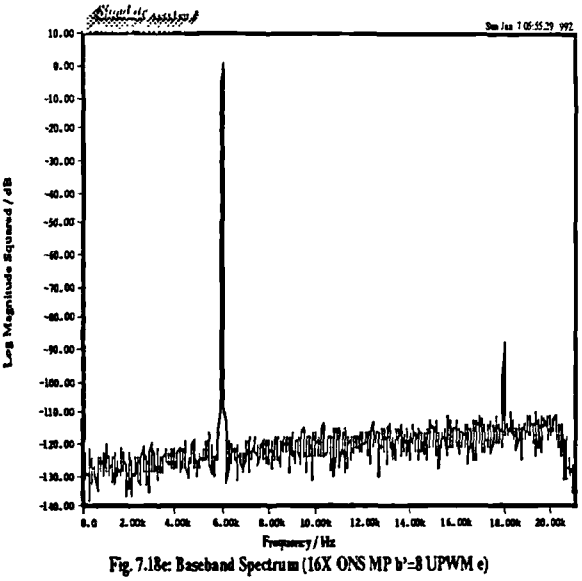


Fig. 7.18e: Baseband Spectrum (16X ONS MP b'=8 UPWM e)

in the high frequency noise power for the alternate asymmetric case as compared with the fixed asymmetric result. As we stated earlier, more research is necessary to provide accurate explanations of these effects.

In any case, for both standard and minimum phase NTF systems oversampling further reduces the effect by again providing a larger bandwidth over which the errors are spread. This is shown in Figs. 7.17c and 7.18c and Figs. 7.17d and 7.18d for fixed asymmetric and alternative asymmetric modulation, respectively. For completeness the standard and minimum phase NTF double sided symmetric system results are shown in Figs. 7.17b and 7.18b. As in the eight times oversampling case, there is no obvious increase in baseband noise power.

The situation is marginally better for two sample consecutive UPWM. This system is thought to suffer from a combination of problems. To understand what is believed to be happening we back up slightly and begin by considering the trailing edge waveforms of Fig. 7.19a-b. The first waveform represents a PWM signal generated by a finely quantized, non-noise shaped input signal. The second waveform represents the output of a system which uses a noise shaper. The noise shaper lengthens or shortens the width of the pulses in a special way such that high quality baseband performance can be achieved. The lengthening or shortening of a particular pulse depends on how the widths of the previous pulses were altered—particularly on how the the immediately previous pulse was altered. Next consider two sample consecutive waveforms arising from finely quantized systems without noise shaping and coarsely quantized systems with noise shaping. These are shown in Fig. 7.20a-b, respectively. In each case we can consider a two sample consecutive PWM pulse as being comprised of leading and trailing edge "sub-pulses" about the center of the pulse interval. Noise shaping does vary the width of each sub-pulse, however, because the trailing edge of the first sub-pulse is the same as the leading edge of the second sub-pulse (i.e., the sub-pulses are connected to form a single pulse), the width of the *overall* pulse is not varied properly. In fact, the tendency of the noise shaper to produce adjacent output values which are alternately too big and then too small seems to have the effect of varying the *position* of the pulse more than its width [Hi92f].

As stated in Chapter Four the solution is to decorrelate the error components associated with the noise shaper output samples that modulate the leading edge of the pulse from the error components of the samples modulating the the trailing edges. For white noise shaper requantization error, this is done by interleaving zeros between the coefficients of the impulse response of the noise shaper NTF. Doing so for the fifth order standard and minimum phase NTFs results in the new symmetric type NTFs shown in Fig. 7.21a-b. Repeating the eight bit two sample consecutive tests with these NTFs yields the output

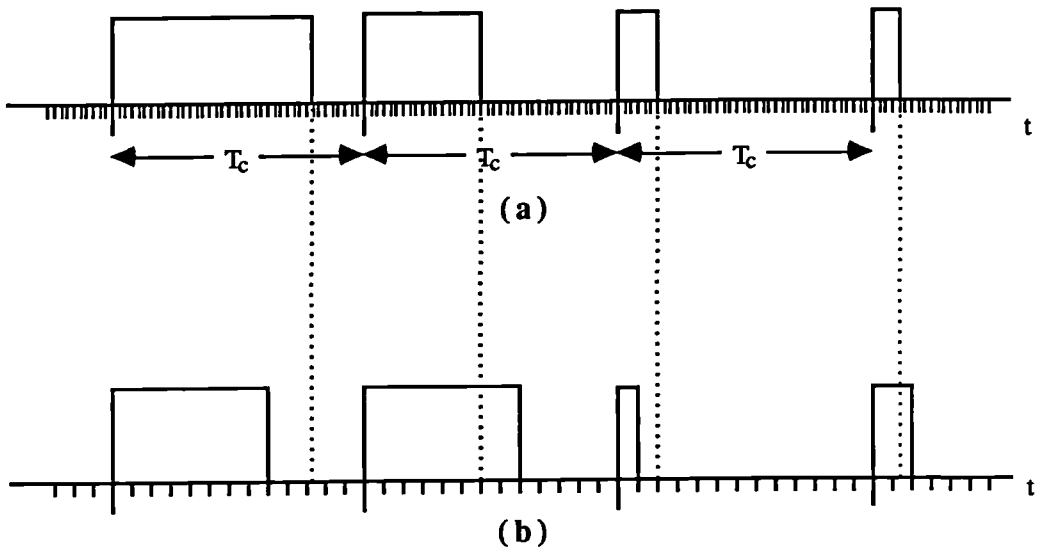


Fig. 7.19: UPWM and ONS/UPWM Trailing Edge Waveforms

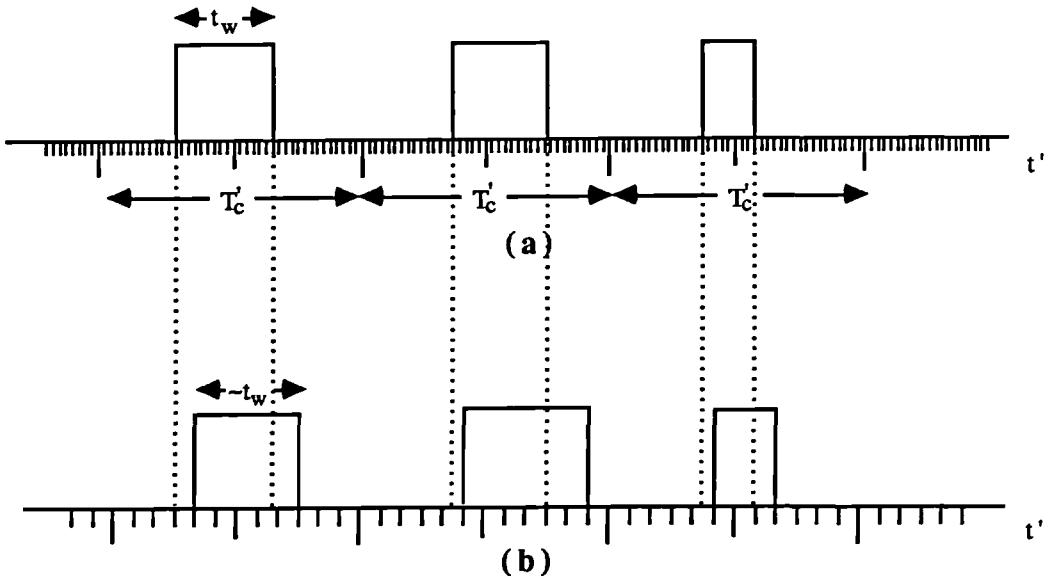


Fig. 7.20: UPWM and ONS/UPWM Two Sample Consecutive Waveforms

spectra shown in Fig. 7.22a-b. The improvements are dramatic with no noticeable increase in noise power for the minimum phase NTF case. For the standard NTF case the small increase in noise power indicates that an additional effect is at work. This is thought to again be a more mild version of the noise foldback/intermodulation problem we encountered for trailing edge modulation. (In terms of the tone analogy, two sample consecutive foldback and intermodulation distortion is lower than that of single sided modulation but higher than that of double sided modulation.) As with the trailing edge case the standard NTF systems, which generate much more high frequency noise power than their minimum phase counterparts, suffer from higher levels of baseband noise power.

For completeness, Figs. 7.17e and 7.18e (without zero interleaving) show that as before 16 times oversampling reduces the noise power problem for both the standard and minimum phase NTF systems. Also, out of interest, Fig. 7.23a shows the output of a seven bit two sample consecutive system with a minimum phase zero interleaved NTF. Before zero-interleaving the order of the NTF is 14. Its frequency response is shown in Fig. 7.23b. At the expense of increased feedback filter complexity it avoids all of the unwanted effects encountered earlier and operates at a reduced modulator clock speed—the same as the eight bit, eight times oversampling trailing edge system ($\sim 90.3MHz$).

Before leaving this subsection we present one last set of results which indicate the presence of yet another undesirable noise effect. This effect is of particular relevance for some of the results presented in the next section. Let us consider the eight bit ONS/trailing edge systems. The use of a minimum phase NTF for this system was shown to virtually eliminate the increased baseband noise power observed earlier. However, upon application of a large 20.001kHz input to this system a slight increase in the high frequency (near 20kHz) baseband noise power is observed for the trailing edge case. This is shown in Figs. 7.24a along with zoomed-in plot of the noise floor in Fig. 7.24b. (For comparison, we include the output of the standard NTF system for the same input in Fig. 7.25a.) This small effect (of Fig. 7.24a-b) is believed to be due to intermodulation between the tone input and the noise shaper output noise in the region of 20-40kHz. In fact, this effect can be made much more dramatic for NTFs with steep transition bands directly above the baseband. Fig. 7.26a show the frequency response of such an NTF (with very high order) designed for use in an eight bit, eight times oversampling system. The output of the trailing edge system using this NTF with a 20.001kHz, M=0.8 input is shown in Fig. 7.26b. The very large increase in baseband noise power is obvious. As shown in Fig. 7.26c the effect is less severe for a lower frequency input.

Next, we consider an NTF with an attenuation band increased to 40kHz. The frequency response is shown in Fig. 7.27a. The output of the trailing edge system using this

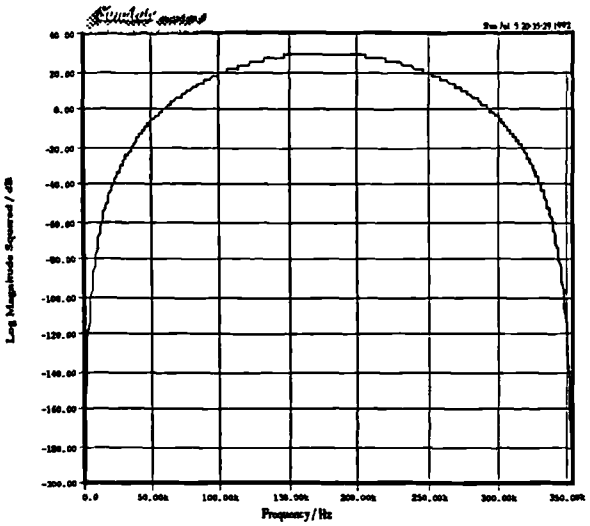


Fig. 7.21a: Zero Interleaved NTF (Standard)

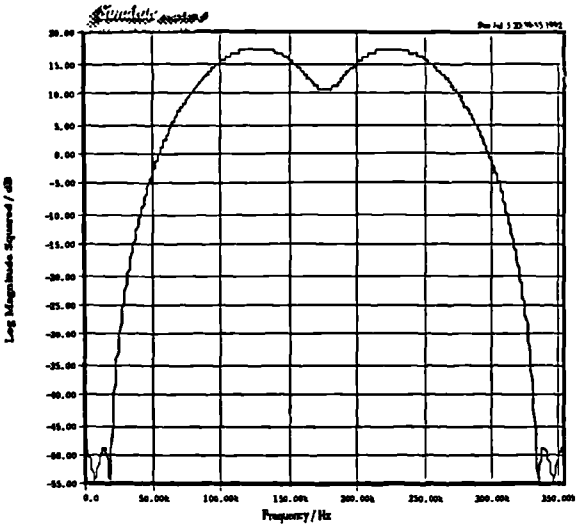


Fig. 7.21b: Zero Interleaved NTF (MP)

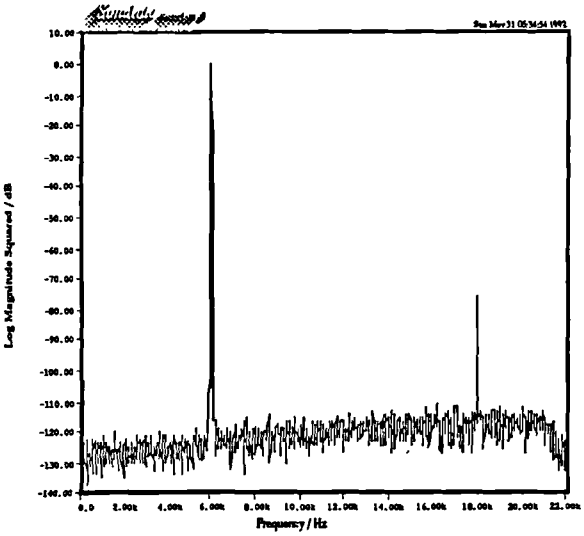


Fig. 7.22a: Baseband Spectrum (ONS Standard ZI b'=8 UPWM e)

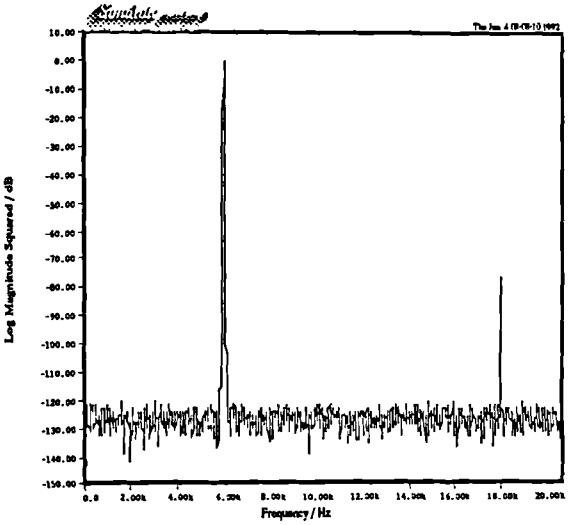


Fig. 7.22b: Baseband Spectrum (ONS MP ZI b'=8 UPWM e)

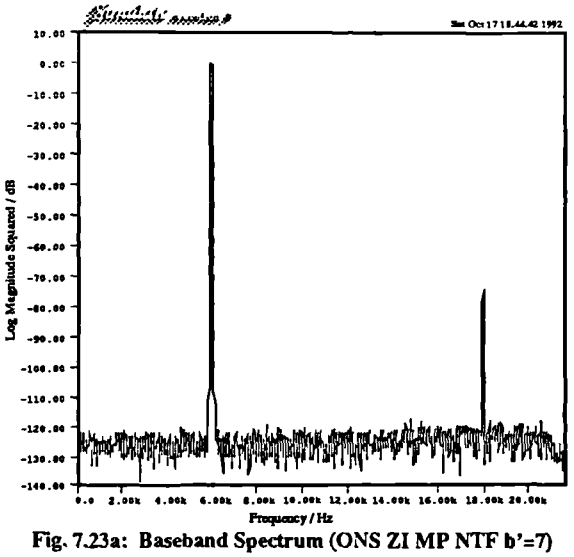


Fig. 7.23a: Baseband Spectrum (ONS ZI MP NTF b'=7)

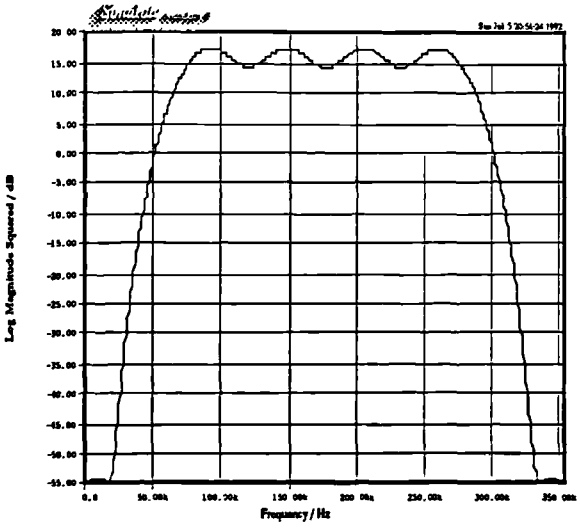


Fig. 7.23b: Zero Interleaved NTF (b'=7 MP N=14 before interleaving)

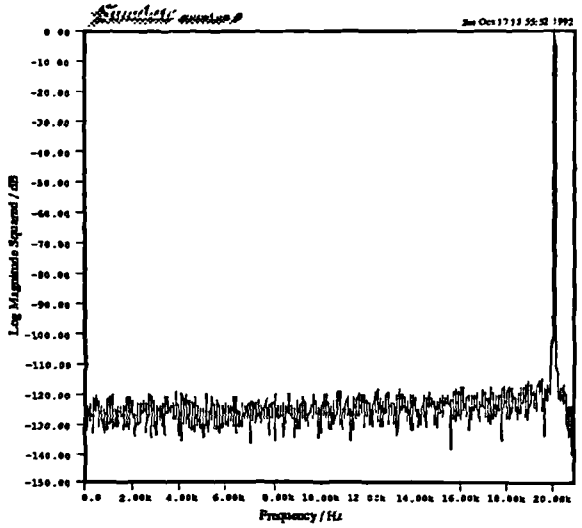


Fig. 7.24a: Spectrum for High Freq. Input (ONS MP b'=8 UPWM a)

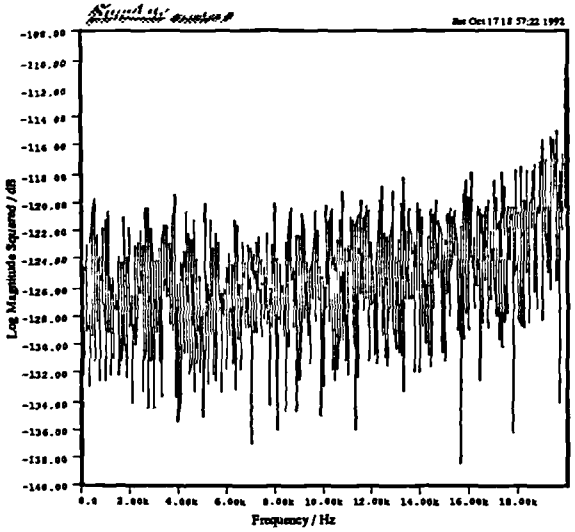


Fig. 7.24b: Noise Spectrum (ONS MP b'=8 UPWM a)

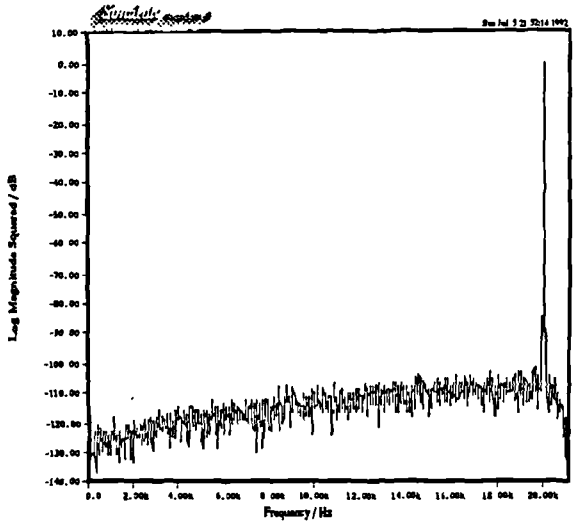


Fig. 7.25: Output Spectrum for High Frequency Input (ONS Standard b'=8 UPWM a)

NTF is shown in Fig. 7.27b. The figure indicates a significant improvement with this NTF—this in spite of the increased high frequency noise power it produces (compared to the 20kHz attenuation band NTF of Fig. 7.26a). The improved performance is due to the fact that much of the noise which could intermodulate with the signal to produce a baseband effect (i.e., noise in the 20-40kHz band) has been shifted to higher frequency regions.

Lastly, we extend the NTF attenuation band to 60kHz. The frequency response of this NTF is shown in Fig. 7.28a. Note that the larger high frequency gain is nearer to the peak gain of the standard fifth order NTF. The output of this system is shown in Fig. 7.28b. We see that the high frequency noise power gain is large enough to again cause the

noise intermodulation/foldback effect we initially observed in the fifth order standard NTF case.

We stress that the results lead us to believe that the above effect is not just a manifestation of the noise intermodulation/foldback problem described earlier. It is unlikely that the undesirable noise effects associated with the standard NTF trailing edge systems (Fig. 7.15a and 7.25) are due to the noise shaper noise intermodulating with the signal. This is because inspection of the frequency responses of the standard and minimum phase fifth order NTFs actually shows that, in spite of yielding much better results, the noise shaper output noise power is in fact *larger* over the 20-40kHz band in the minimum phase NTF. (It does not make sense to claim that the signal/noise shaper noise intermodulation effect is principally responsible for the poorer standard NTF performance when the standard NTF actually produces *less* noise in the 20-40kHz band than the minimum phase system, which gives better results.) Moreover, it is also unlikely that the problem just observed in Fig. 7.26 stems from the noise foldback/intermodulation distortion effect described earlier. (This is for several reasons. Figs. 7.26b and 7.26c indicate a strong dependence of the noise on input signal frequency which was not observed in the fifth order standard NTF cases (Figs. 7.14a and 7.25) where noise foldback/intermodulation was believed to be responsible for the poor results. Also, for the high order NTFs, widening the attenuation band from 20kHz to 40kHz such that no baseband noise/signal intermodulation was possible yielded greatly improved results but at the same time raised high frequency noise power considerably. This further leads us to believe that signal/noise intermodulation as opposed to noise foldback/intermodulation was the principle effect in the results of Fig. 7.26.)

Fortunately, in our fifth order minimum phase NTF the signal/noise intermodulation is very small. However, this effect should be taken into account when designing the shape of the NTF—particularly for low oversampling, low output wordlength, and/or higher than 16 bit quality applications.

7.3.3 Overview

At this stage we sum up by stating that ONS networks can be used in conjunction with digital UPWM systems. However, careful selection of the noise shaper NTF is necessary to ensure levels of performance commensurate with those of 16 bit systems which do not use noise shaping. We observed noise problems in some of the single sided, double sided, and two sample consecutive systems when preceded by certain noise shaping

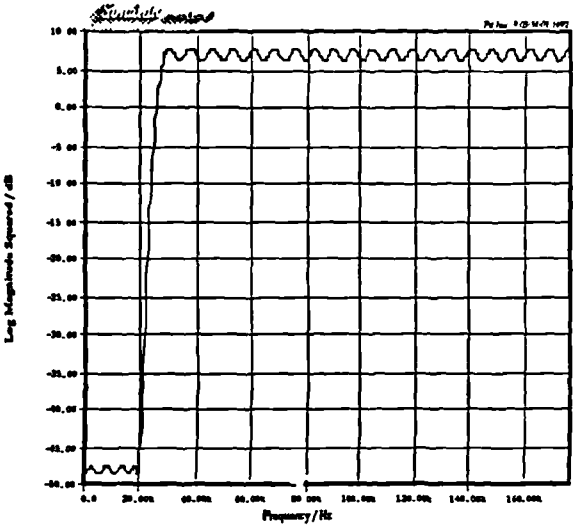


Fig. 7.26a: High Order NTF (20kHz attenuation band)

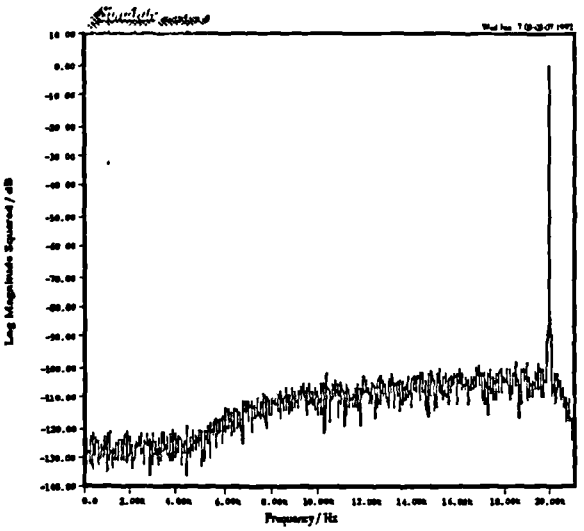


Fig. 7.26b: Baseband Output Spectrum

Fig. 7.26c (overleaf)

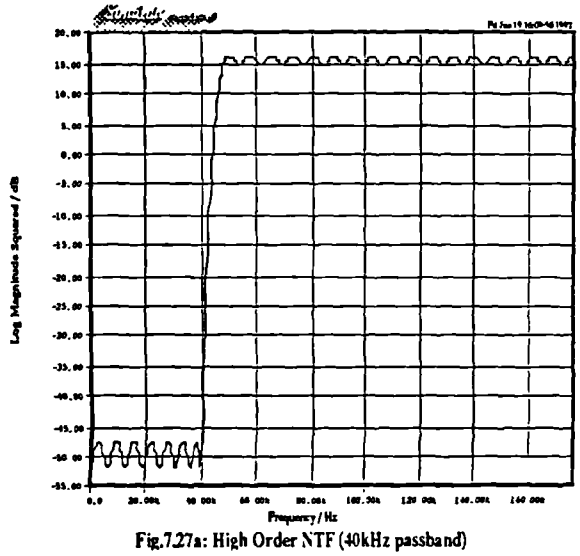


Fig.7.27a: High Order NTF (40kHz passband)

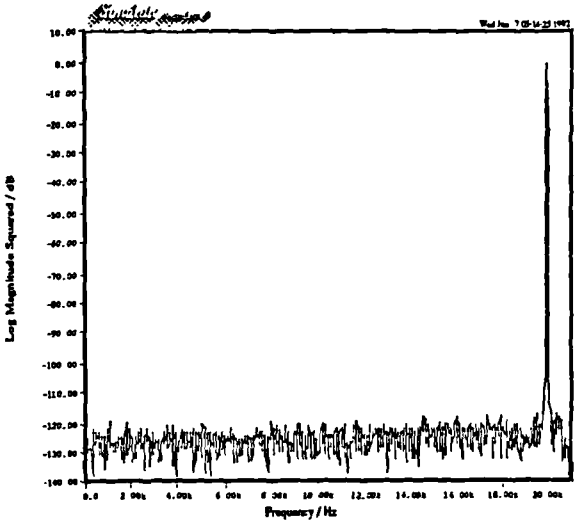


Fig. 7.27b: Baseband Output Spectrum

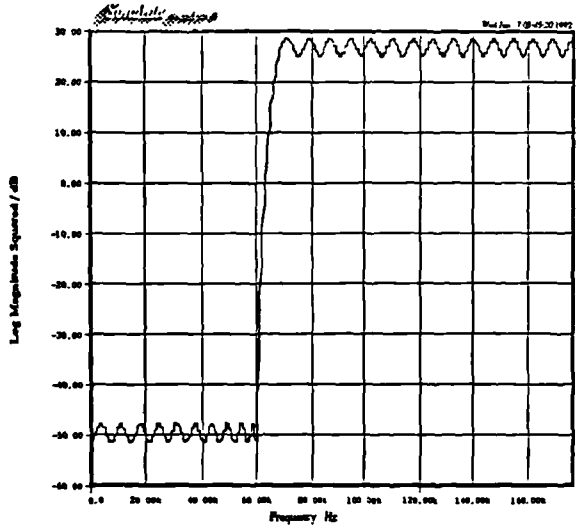


Fig. 7.28a: High Order NTF (60kHz attenuation band)

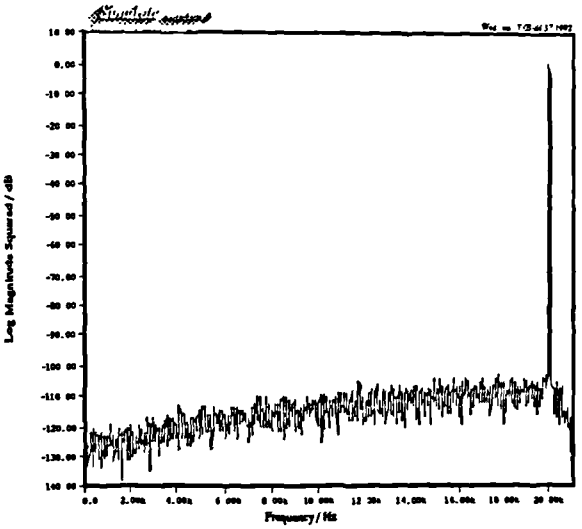


Fig. 7.28b: Baseband Output Spectrum

networks. However, the use of noise shapers possessing special minimum phase, low noise power gain NTFs improved the performance of the single sided and two sample consecutive systems (with a zero interleaved feedback filter also required in the latter). For the double sided cases, symmetric modulation performed best followed by alternative alternate asymmetric and then fixed asymmetric modulation. Generally, we found that when increased noise power problems existed, additional oversampling helped to reduce the problem.

On the basis of the experiments conducted, it is believed that for f_c fixed, two sample consecutive UPWM, when used in conjunction with a minimum phase NTF and zero interleaved feedback filter coefficients, is the best of the five UPWM modulation types. (However, we do acknowledge having disregarded at this stage potentially important practical issues such as modulator clock speed and computational complexity.) Using this system as a UPWM performance benchmark, we next see how the introduction of a cross point deriver can further improve the performance of ONS/PWM DACs.

7.4 ONS/PNPWM DACs

In this section we present results from the simulation of ONS/trailing edge PNPWM based converters as shown in Fig. 7.29. These systems are designed to improve the performance of the ONS/UPWM DACs by applying a special linearization algorithm to the signal prior to modulation. As explained in Chapter Four such procedures attempt to numerically approximate the natural sampling cross point times of the original analogue signal with the PWM sawtooth comparison waveform. These time values are then applied to the standard UPWM modulator with the goal of achieving very low distortion NPWM performance.

From Fig. 7.29 we see that the cross point derivation stage precedes the ONS stage. The order of these two stages is important. In theory, linearizing a noise shaped version of the input would yield advantages such as the elimination of the signal/noise intermodulation problem observed earlier. (As we shall see PNPWM systems do not exhibit intermodulation distortion.) However, there are some serious disadvantages associated with this approach. In particular, it is much more difficult to form accurate polynomial approximations to the high frequency noise shaper output than to just the highly oversampled, non-noise shaped input signal. Additionally, cross point derivation of the noise shaped signal may well lead to higher levels of baseband noise foldback distortion since, in terms of the tone analogy, NPWM foldback distortion is more severe than that of UPWM. Also, from

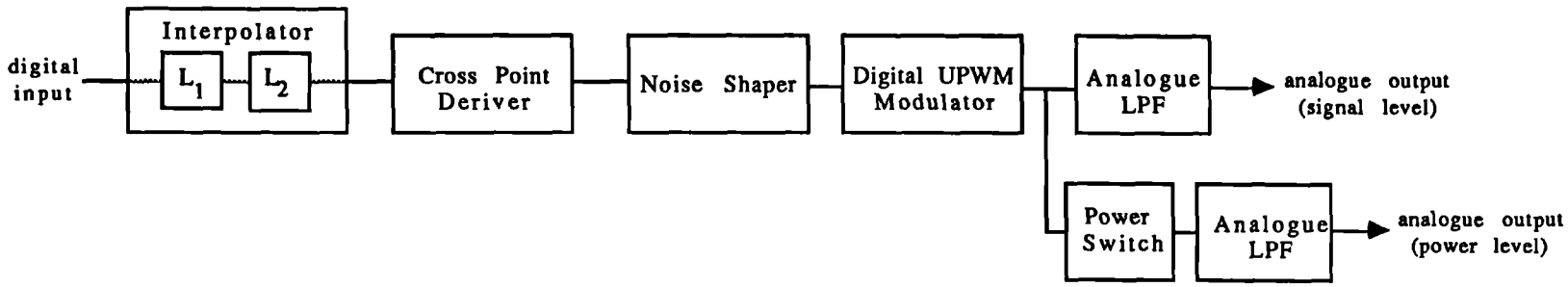


Fig. 7.29: ONS/PNPWM DAC

a more numerical perspective, the floating point numbers generated by the cross point deriver must be converted to integers before being applied to the actual modulator. This happens naturally when the floating point output of the cross point deriver drives the noise shaper. However, if the noise shaper precedes the cross point deriver some additional form of quantization is necessary at the output of the cross point deriver. This can be performed by either a simple rounding operation (which is inadvisable since large new quantization errors would result) or, more realistically, by an additional stage of noise shaping. Overall, it is believed that the relative lack of disadvantages associated with the configuration shown in Fig. 7.29 make it the best choice.

We now investigate the single tone and twin tone performance of ONS/PNPWM DACs with several cross point derivation algorithms of varying complexity. These systems will be compared to the ONS/two sample consecutive UPWM DAC with zero-interleaved NTF.

7.4.1 Plots and Comparisons

Before examining converters with real cross point derivation algorithms we first consider a few systems driven by artificially created "perfect" cross point signals. These signals are produced by a searching procedure using direct calls to the sine function.* We begin with the M=0.8, f_v=20.001kHz cross point derived input signal driving trailing edge modulators with pulse repetition frequencies of 44.1kHz and 352.8kHz as well as an eight bit f_c=352.8kHz ONS/PNPWM system with the fifth order minimum phase NTF. Wideband results are shown for these three systems in Fig. 7.30a-c, respectively. With f_c=44.1kHz the output is messy looking with a large number of tones distributed in a relatively unstructured looking way. When the pulse repetition frequency is raised to f_c=352.8kHz we observe somewhat more familiar looking spectra (both with and without noise shaping) with components located at the carrier and its harmonics as well as at multiples of the input tone about the carrier and its harmonics. The noise floor of the system which uses noise shaping strongly resembles that of the corresponding ONS/trailing edge UPWM wideband spectra of Fig. 7.10a of the previous section. In contrast to the UPWM plots we note the absence of harmonic distortion in the two f_c=352.8kHz plots.

Baseband versions of the two plots for the systems which do not use noise shaping are shown in Fig. 7.31a-b. We see that the performance of the system operating at

* The perfect cross point deriver is not feasible for use in a real system (see Chapter Six).

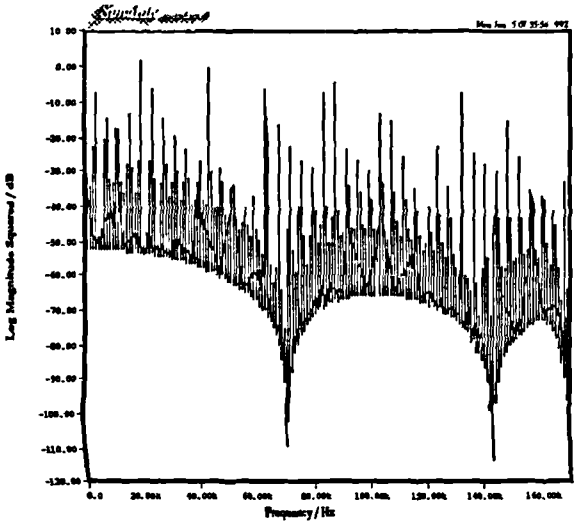


Fig. 7.30a: Wideband Spectrum (PNPWM f)

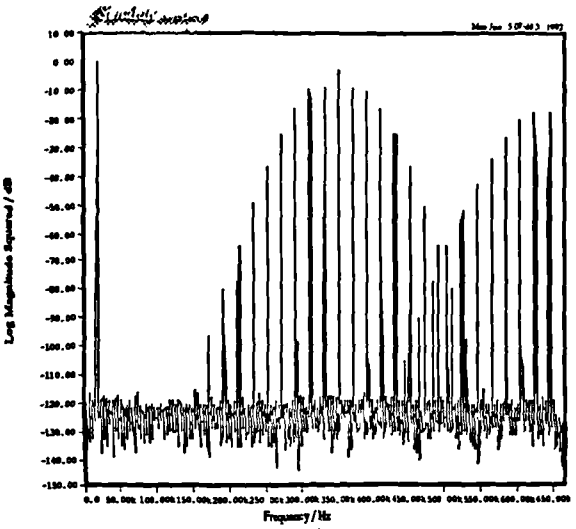


Fig. 7.30b: Wideband Spectrum (w/8X oversampling PNPWM f)

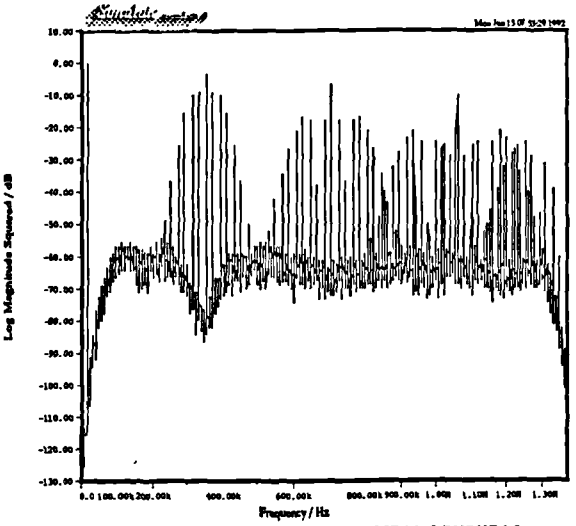


Fig. 7.30c: Wideband Spectrum (8X ONS MP b'=8 PNPWM f)

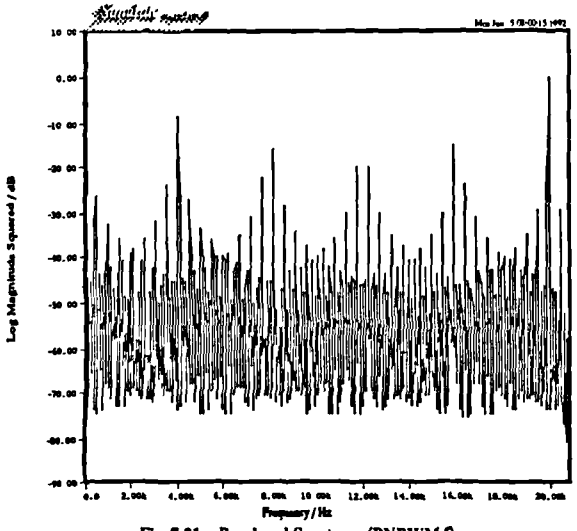


Fig. 7.31a: Baseband Spectrum (PNPWM f)

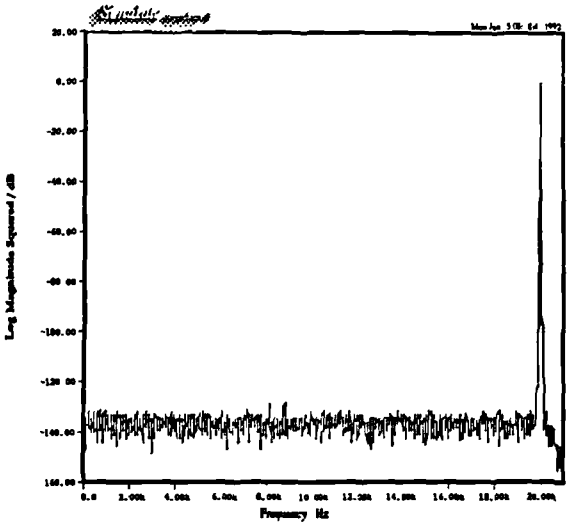


Fig. 7.31b: Baseband Spectrum (w 8X oversampling PNPWM f)

f_c=44.1kHz is very poor indeed with many unwanted tones over the entire baseband. These tones are foldback distortion about the carrier and its harmonics. We see particularly large components folding back from the carrier itself at $|1 \cdot 44.1kHz - 2 \cdot 20.001kHz| \approx 4.1kHz$ and $|1 \cdot 44.1kHz - 3 \cdot 20.001kHz| \approx 15.9kHz$. The other terms are foldback components from higher harmonics of the carrier. The poor performance of this system is not surprising given the theoretical considerations of Chapter Two which predict severe foldback for NPWM modulators with no oversampling. By contrast the system operating at f_c=352.8kHz yields very clean looking results. This is because the large distance between the baseband and the carrier ensures that all the baseband foldback distortion terms have fallen to levels well beneath the noise floor.

Table 7.8 shows how trailing edge PNPWM baseband foldback distortion varies as a function of pulse repetition frequency for a 20.001kHz, M=0.8 sinusoidal input. For comparison, we have also included the UPWM trailing edge results presented earlier. As in the UPWM case, PNPWM foldback distortion decreases as the pulse repetition frequency is increased but does so more slowly, not completely disappearing beneath the noise floor until f_c=352.8kHz.

Table 7.9 shows how the foldback distortion varies as a function of M for f_c=44.1kHz and f_v=20.001kHz. Again the trailing edge UPWM results are included for comparison, and again we see that the distortion figures are lower than those of trailing edge PNPWM. As before, these differences are expected from theoretical considerations discussed in Chapter Two.

Next, we consider an example to show how closely the simulation results correspond to those predicted by theory (i.e., by evaluation of the tone spectra expression and their approximations). In Table 7.10 we look at the second and third multiples of the 20.001kHz input tone folding back from a carrier of 44.1kHz. There is close correspondence between the simulation results and those obtained from direct evaluation of the appropriate terms in the NPWM tone spectrum equation. However, there are sizable differences between these figures and those obtained by evaluating the relevant approximation expression in Table 2.4 of Chapter Two. As earlier, the approximations are poor because the arguments of the associated Bessel functions in the tone spectra equation are not small enough compared to the (low) order of these Bessel functions.

Next, we examine the baseband performance of a variety of ONS/PNPWM DACs under various inputs and compare the results with those obtained by the better UPWM systems. Specifically, we consider ONS/trailing edge PNPWM systems using a "perfect" cross point deriver in addition to those with the first order, third order, and fifth order cross point derivation procedures described in Chapter Five. These will be compared to the

Table 7.8: PNPWM Baseband Foldback Distortion as a Function of f_c

($f_v = 20.001kHz$, $M = 0.800$, $b = 16$)

f_c (L)	Distortion $20\log_{10}(\cdot)$	Modulation Type	
		f	a
44.1kHz (1)	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-8.85	-22.80
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-14.92	-27.90
88.2kHz (2)	$\left\|\frac{F_{1-4}}{F_1}\right\|$	-24.82	-83.05
	$\left\|\frac{F_{1-5}}{F_1}\right\|$	-36.08	-102.37
176.4kHz (4)	$\left\|\frac{F_{1-8}}{F_1}\right\|$	-79.70	•
	$\left\|\frac{F_{1-9}}{F_1}\right\|$	-96.17	•
352.8kHz (8)	$\left\|\frac{F_{1-17}}{F_1}\right\|$	•	•
	$\left\|\frac{F_{1-18}}{F_1}\right\|$	•	•

• Distortion does not exist or is beneath 16 bit noise floor

ONS/two sample consecutive UPWM system. The output spectra of these five systems for an f_v=1.001kHz, M=0.8 input are shown in Fig. 7.32a-e, respectively. As expected the perfect cross point procedure results in no harmonic or foldback distortion. By contrast, the first order result indicates the presence of distortion terms at the third harmonic of the input. However, both the third order and fifth order procedures indicate harmonic distortion free performance. Lastly, the two sample consecutive system yields only third order harmonic distortion (as expected) at a level quite similar to that of first order PNPWM.

Table 7.9: PNPWM Baseband Foldback Distortion as a Function of M

($f_c = 44.1kHz$, $f_v = 20.001kHz$, $b = 16$)

M	Distortion $20\log_{10}(\cdot)$	Modulation Type	
		f	a
0.025	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-34.14	-54.29
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-71.39	•
0.050	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-28.13	-48.33
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-59.35	-77.03
0.100	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-22.17	-42.24
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-47.34	-65.06
0.200	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-16.36	-31.16
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-35.46	-52.93
0.400	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-11.22	-29.88
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-24.07	-40.70
0.800	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-8.85	-22.80
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-14.92	-27.90

• Distortion does not exist or is beneath 16 bit noise floor

Table 7.10: Correspondence with Theory (PNPWM Foldback Distortion)
($f_c = 44.1kHz$, $f_v = 20.001kHz$, $M = 0.800$, $b = 16$)

Source	Distortion $20\log_{10}(\cdot)$	Modulation Type f
Theoretical	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-8.96
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-15.17
Approximation	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-3.68
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-11.35
Simulation	$\left\|\frac{F_{1-2}}{F_1}\right\|$	-8.85
	$\left\|\frac{F_{1-3}}{F_1}\right\|$	-14.92

We also note that the rise in the noise floor of the output of the system with the perfect cross point deriver is due the noise effects described in the previous section. As this algorithm produces an output which is effectively sampled at the (high) pulse repetition frequency its noise floor is lower than at the outputs of the other more realistic cross point derivers which accept data interpolated up from the Nyquist rate. The increase in output baseband noise in the perfect cross point deriver case is visible because of this lower noise floor. In the more realistic systems the increases in noise power are very small compared to the higher levels of inherent quantization noise, and hence such effects are less noticeable.

The tests are repeated for a 6.001kHz, M=0.8 input. The corresponding results are shown in Fig. 7.33a-e. Again no distortion can be seen in the perfect, the third order, or the fifth order PNPWM results. However, we observe distortion at the second and third harmonics of the input for the first order PNPWM output as well as third harmonic

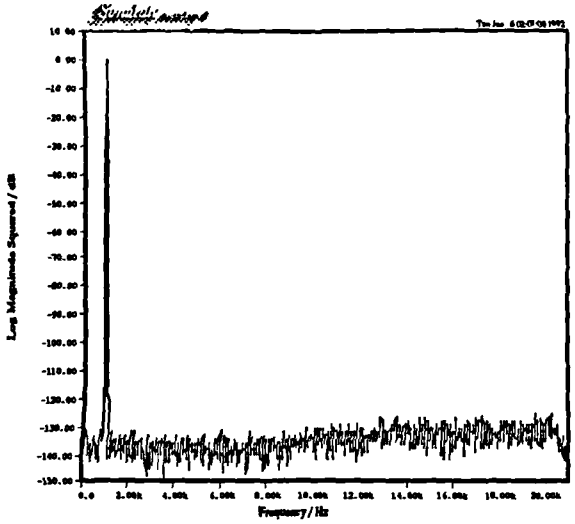


Fig. 7.32a: Baseband Spectrum (ONS MP b'=8 PNPWM f)

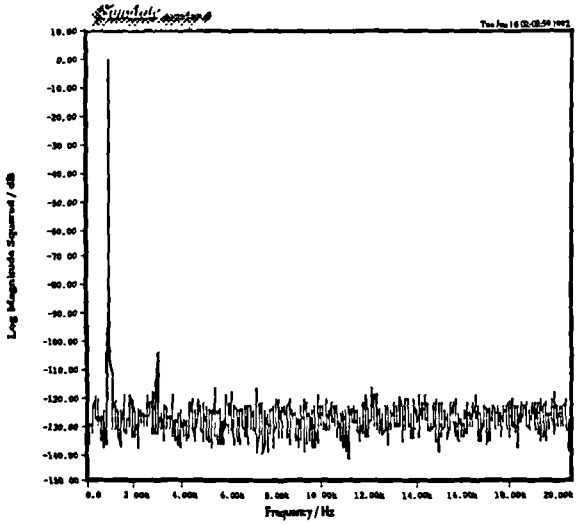


Fig. 7.32b: Baseband Spectrum (ONS MP b'=8 PNPWM h)

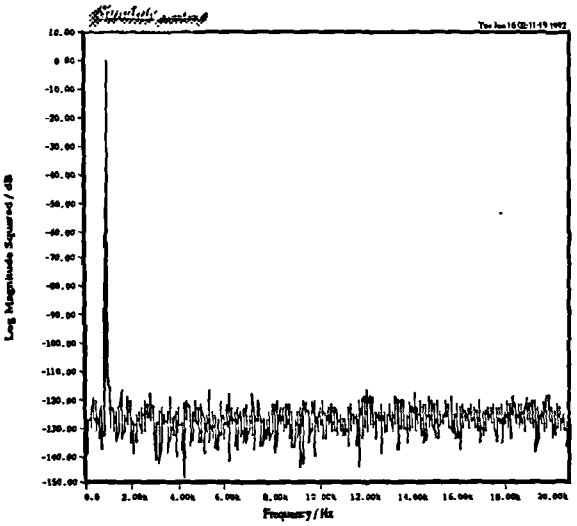


Fig. 7.32c: Baseband Spectrum (ONS MP b'=8 PNPWM i)

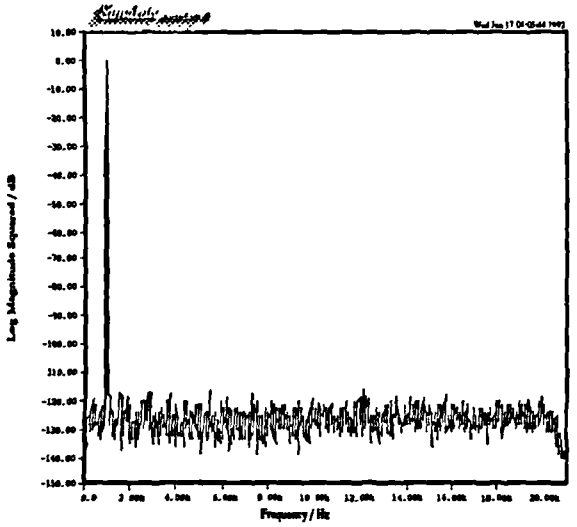


Fig. 7.32d: Baseband Spectrum (ONS MP b'=8 PNWPM j)

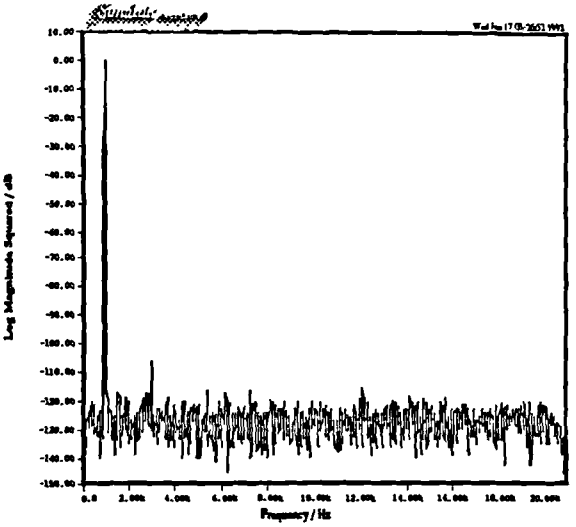


Fig. 7.32e: Baseband Spectrum (ONS MP ZI b'=8 UPWM e)

distortion in the two sample consecutive UPWM result. As before, the level of the third harmonic in both these results is similar. It is interesting to note that in contrast to ordinary trailing edge UPWM, first order PNPWM produces second harmonic distortion which is actually smaller than the third harmonic distortion.

Lastly, results for an f_v=20.001kHz, M=0.8 sinusoidal input are shown in Fig. 7.34a-e. The increase in high frequency baseband noise power for all the trailing edge systems is expected (see Figs. 7.24a-b from the previous sub-sections) and should not be thought of as being caused by the cross point derivation algorithms themselves. Once again no distortion is evident in the perfect cross point deriver case. The first order algorithm does not appear to generate any baseband distortion. This is because the high frequency input is such that all the harmonic distortion lies outside the baseband. (Most certainly, however, there are magnitude and/or phase errors on the tone itself.) Interestingly, it is seen that the third order procedure does result in a small term at approximately 17.5kHz. This is eliminated in the fifth order cross point system. Lastly, in a similar way to the first order PNPWM system, no baseband distortion is seen in the two sample consecutive UPWM output.

Next, we investigate the performance of the above systems with twin tone inputs. The first test consists of the application of a pair of sinusoids at f_{v_1}=0.251kHz, M_1=0.78 and f_{v_2}=8.001kHz, M_2=0.195. (The modulation depths have been lowered slightly from those in Section 7.2 to prevent overloading the quantizer inside the noise shaper.) The results for the five systems are shown in Fig. 7.35a-e. We see that the perfect cross point deriver system reproduces the two tones with no harmonic or intermodulation distortion. In contrast, the PNPWM system with first order cross point deriver generates rather large intermodulation terms about the 8.001kHz tone and its second harmonic, 16.002kHz. These problems are seen to be eliminated by the systems using the third order and the fifth order cross point algorithms. The two sample consecutive UPWM system exhibits distortion about 8.001kHz and 16.002kHz but at levels slightly lower than those in the first order PNPWM system.

The second intermodulation test input consists of two tones at f_{v_1}=11.001kHz, f_{v_2}=12.001kHz with M_1=M_2=0.49. (Again the modulation depths have been lowered slightly to avoid overloading the noise shaper's quantizer.) The results are shown in Fig. 7.36a-e. As before, no distortion is evident in the perfect cross point deriver system. In the first order system, however, we see large intermodulation distortion components at approximately 10kHz and 13kHz with smaller terms at roughly 1kHz and 2kHz. Also, the third order cross point algorithm system produces smaller distortion terms again

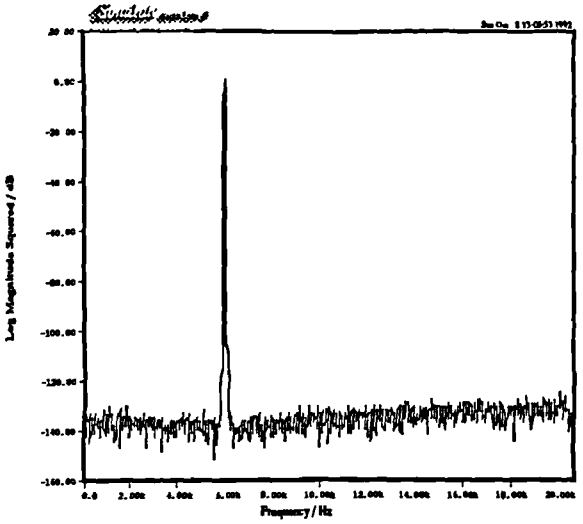


Fig. 33a: Baseband Spectrum (ONS MP bb'=8 PNPWM f)

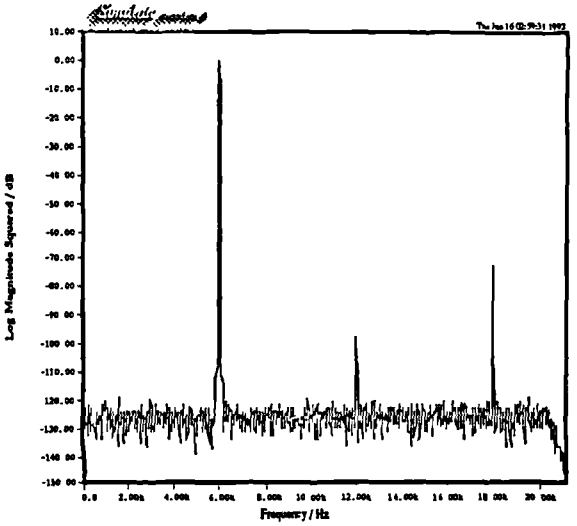


Fig. 7.33b: Baseband Spectrum (ONS MP b'=8 PNPWM h)

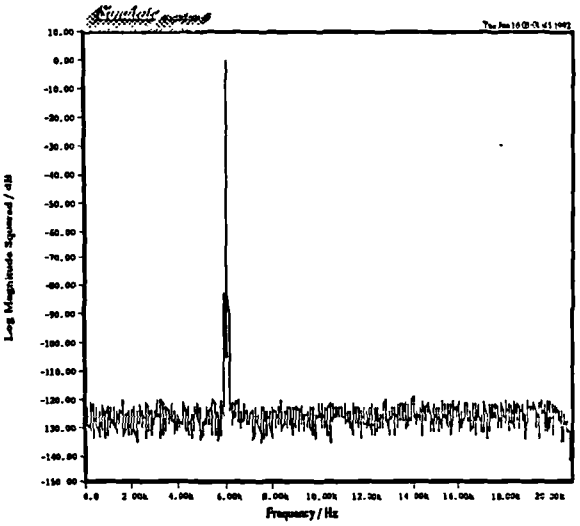


Fig. 7.33c: Baseband Spectrum (ONS MP b'=8 PNPWM i)

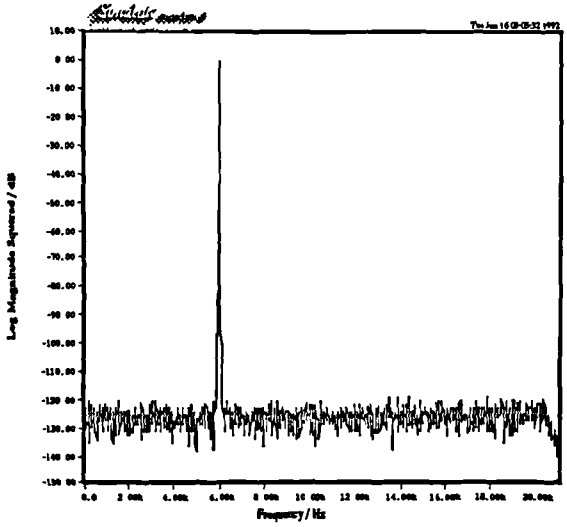


Fig. 7.33d: Baseband Spectrum (ONS MP b'=8 PNPWM j)

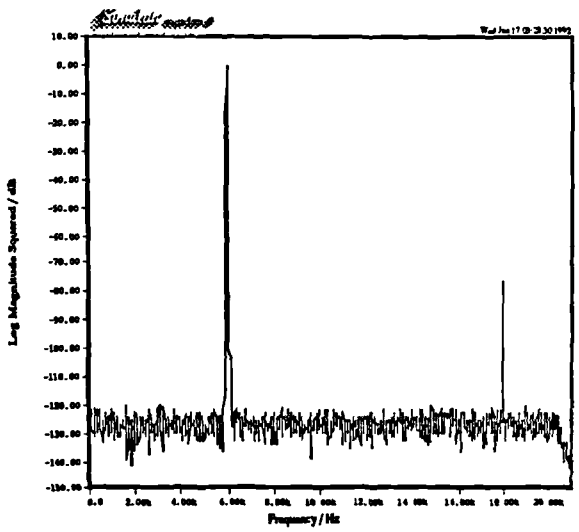


Fig. 7.33e: Baseband Spectrum (ONS MP ZI b'=8 UPWM e)

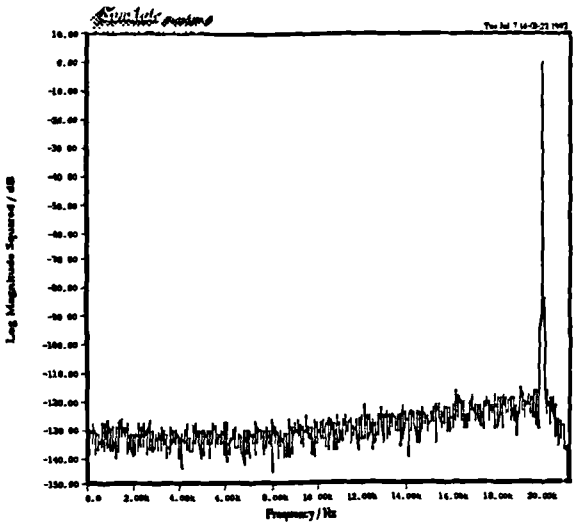


Fig. 7.34a: Baseband Spectrum (ONS MP b'=8 PNPWM f)

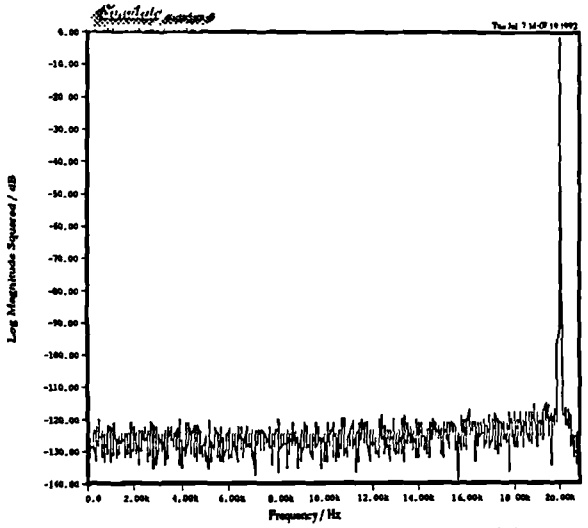


Fig. 7.34b: Baseband Spectrum (ONS MP b'=8 PNPWM h)

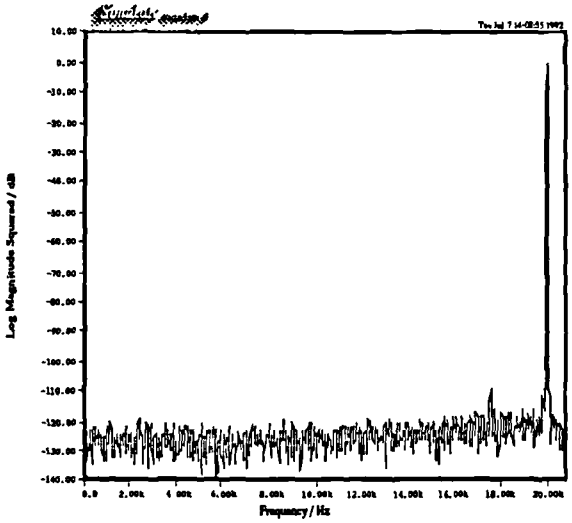


Fig. 7.34c: Baseband Spectrum (ONS MP b'=8 PNPWM f)

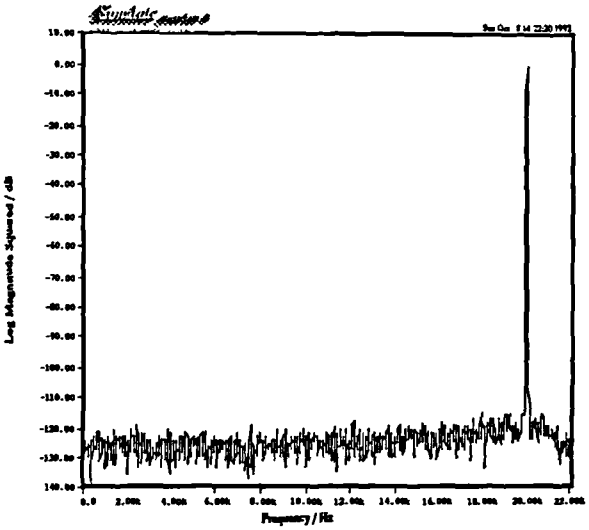


Fig. 7.34d: Baseband Spectrum (ONS MP b'=8 PNPWM j)

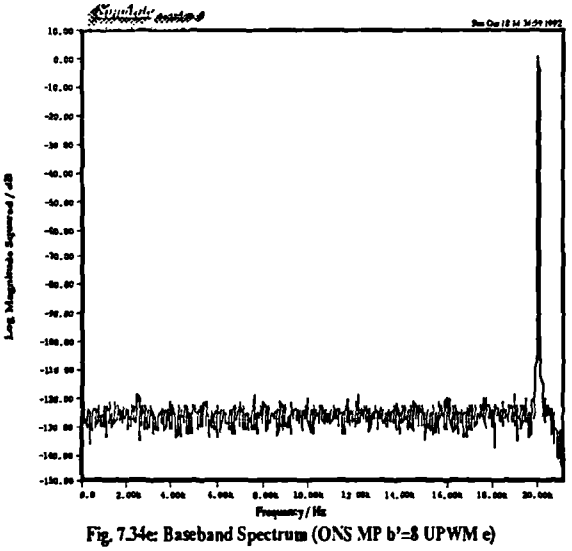


Fig. 7.34e: Baseband Spectrum (ONS MP b'=8 UPWM e)

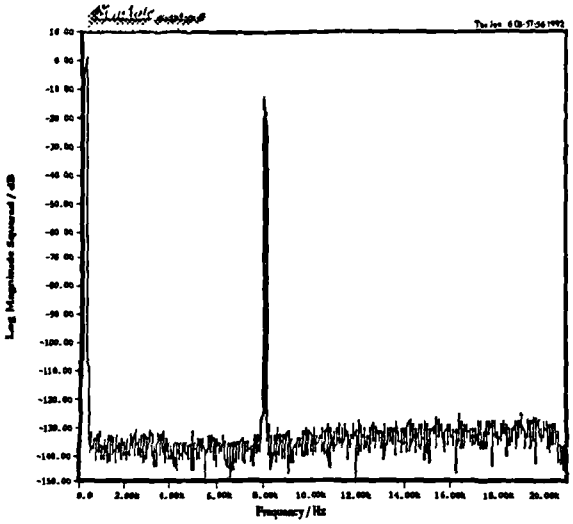


Fig. 7.35a: Intermodulation Test 1 (ONS MP b'=8 PNPWM f)

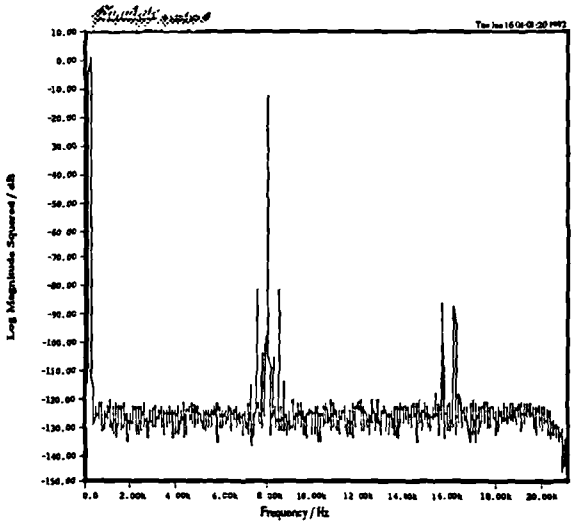


Fig. 7.35b: Intermodulation Test 1 (ONS MP b'=8 PNPWM h)

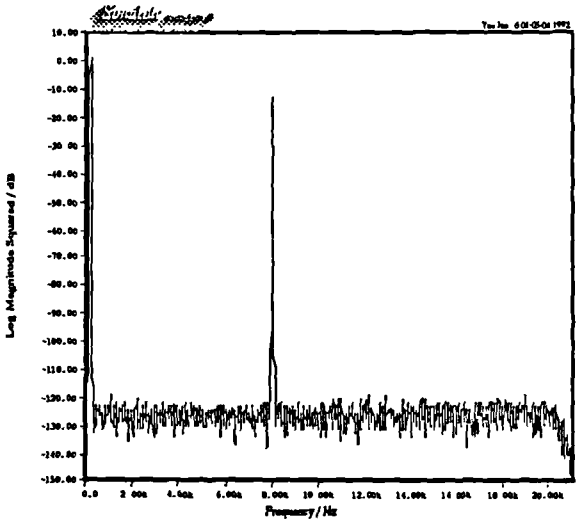


Fig. 7.35c: Intermodulation Test 1 (ONS MP b'=8 PNWPM f)

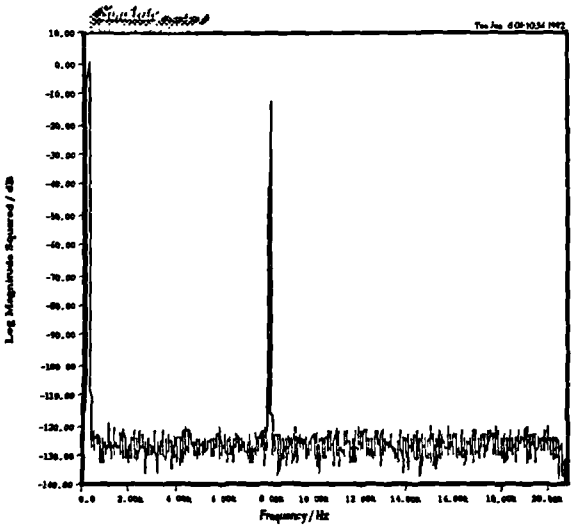


Fig. 7.35d: Intermodulation Test 1 (ONS MP b'=8 PNPWM j)

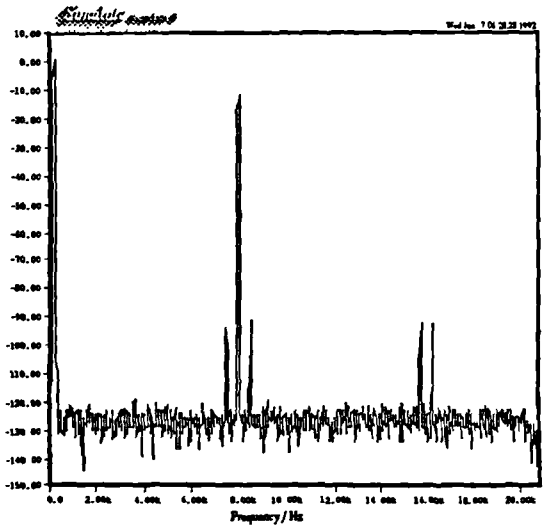


Fig. 7.35e: Intermodulation Test 1 (ONS MP ZI b'=8 UPWM e)

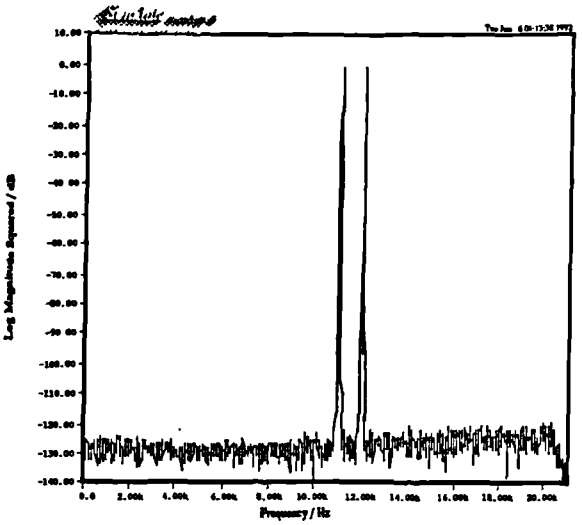


Fig. 7.36a: Intermodulation Test 2 (ONS MP b'=8 PNPWM f)

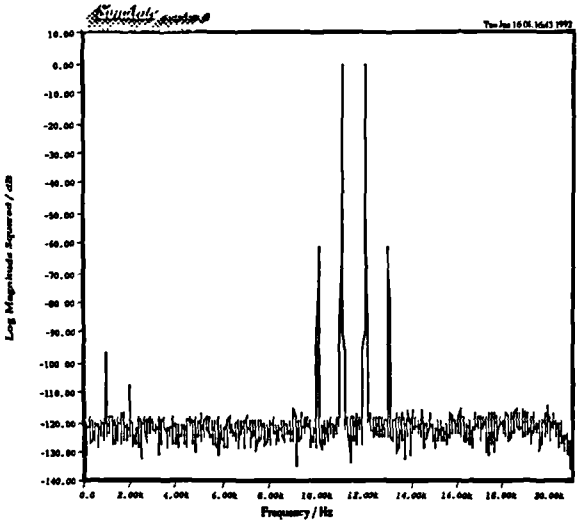


Fig. 7.36b: Intermodulation Test 2 (ONS MP b'=8 PNPWM h)

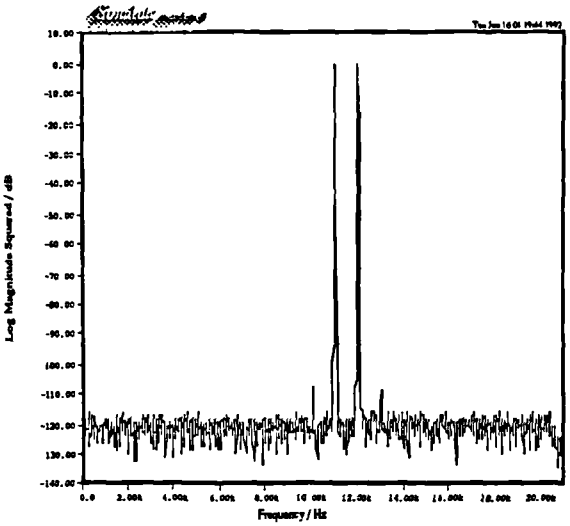


Fig. 7.36c: Intermodulation Test 2 (ONS MP b'=8 PNPWM i)

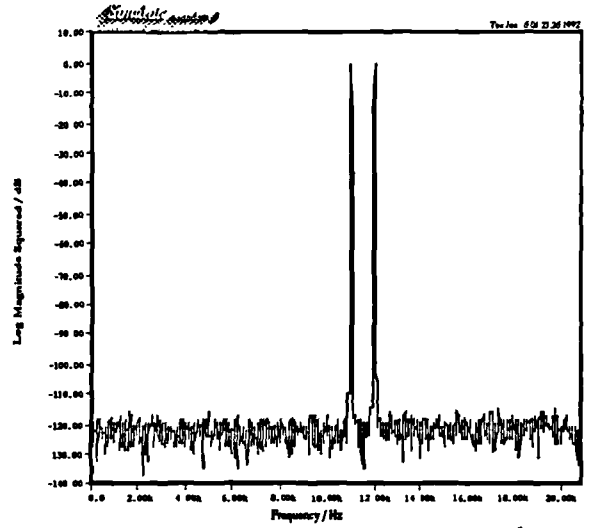


Fig. 7.36d: Intermodulation Test 2 (ONS MP b'=8 PNPWM j)

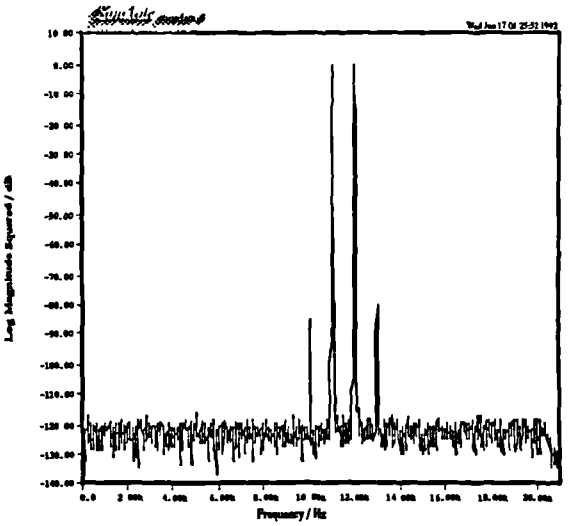


Fig. 7.36e: Intermodulation Test 2 (ONS MP ZI b'=8 UPWM e)

at about 10kHz and 13kHz. These are eliminated in the system using the fifth order cross point derivation algorithm where no distortion is visible. Lastly, the two sample consecutive system is seen to produce distortion terms at approximately 10kHz and 13kHz. We note that these distortion terms are smaller than those generated by the first system but larger than those of the third order system.

It is also interesting for us to compare the SNR of the four realistic systems as a function of input frequency and modulation depth. By "SNR" we mean the ratio of signal power to total error power (including all noise and distortion). See Section 6.9 of Chapter Six for more details. These comparisons are shown in Figs. 7.37a-b, respectively. Beginning with the first order system, Fig. 7.37a indicates that the SNR is highest for low frequency inputs. This is expected since they present less of a demand on the cross point deriver algorithms. However, as the input frequency rises approaching the 20.001kHz worst case input we see that the SNR of the DAC falls dramatically. This is due to a combination of harmonic distortion and amplitude/phase errors on the signal frequency. For input frequencies greater than 10kHz any harmonic distortion would lie outside the baseband. Therefore, in such cases the poor SNR is due exclusively to these amplitude and/or phase errors the DAC produces in its attempt to recreate the input signal.* This effect is not as pronounced in the system using the third order procedure. Here we see a degradation in SNR only for the very high frequency inputs. The DAC with the fifth order procedure exhibits SNR figures which are roughly independent of signal frequency. (In the next section we see that the marginal loss in SNR at 20.001kHz is due to a combination of high frequency noise power shown in Fig. 7.24b and slight errors on the tone itself.) It is also interesting to note that two sample consecutive UPWM exhibits higher SNR figures than first order PNPWM but lower than third order PNPWM. Moreover, the odd looking fluctuation in SNR between 6kHz-8kHz for two sample consecutive UPWM is due to the fact that for input frequencies higher than about 7kHz the SNR measurements are suddenly no longer affected by third order harmonic distortion which is now out-of-band.

In Fig. 7.37b we see how the SNR of these four systems change with modulation depth (i.e., signal amplitude). The signal frequency, 6.001kHz, is chosen so as to allow for both amplitude/phase distortion of the fundamental as well as harmonic distortion to

* This point requires more looking into. Additional research has indicated that none of the cross point derivation procedures actually produce phase distortion in the output at the signal frequency. All output errors on the input signal itself are thought to be due to nonlinear frequency and amplitude dependent scaling errors. In such high resolution systems small scaling errors can have devastating effects on the SNR as defined in Eq. 6.8. However, in subjective terms, such errors would be much less serious, especially when relegated to the very high frequency end of the audio band. Thus, the very low SNR figures associated with the first order procedure should not necessarily be met with despair.

Fig. 7.37.a: SNR as a Function of Signal Frequency

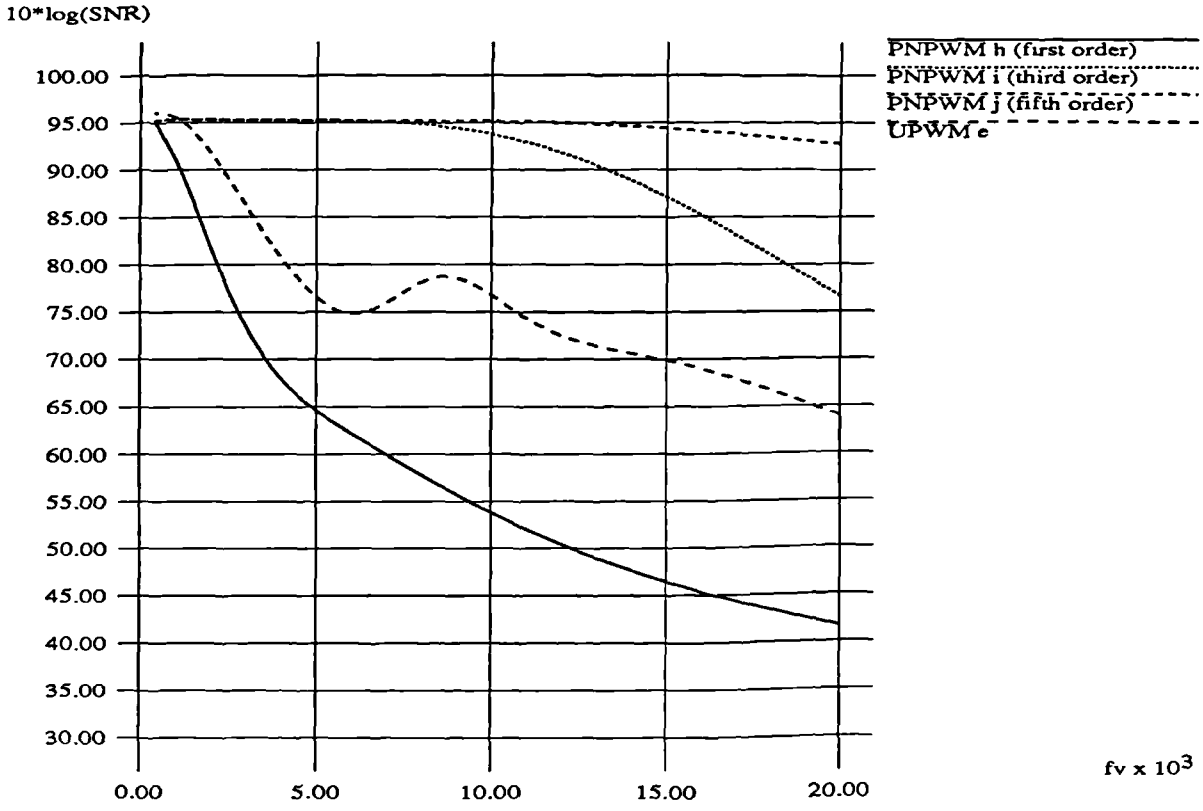
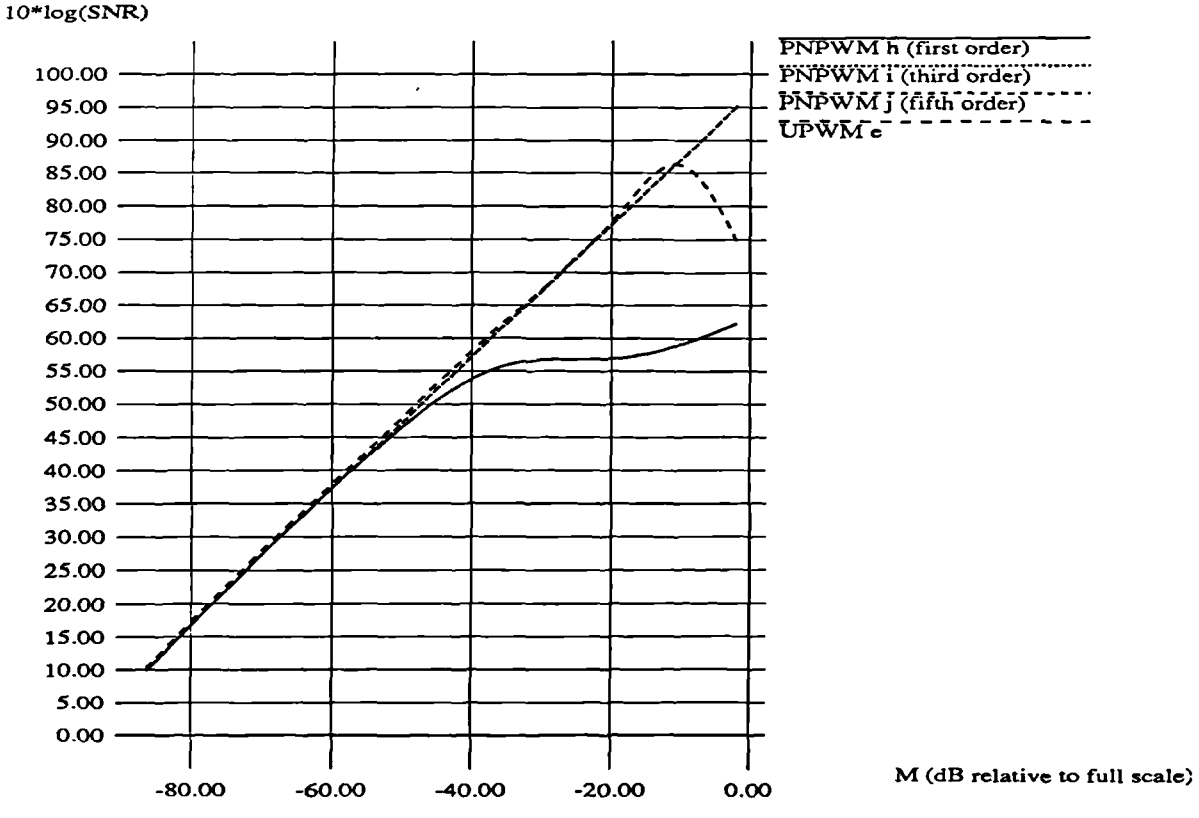


Fig 7.37b: SNR as a Function of Modulation Depth



potentially contribute to DAC errors. The first order procedure performs well for small inputs. However, as the modulation depth is increased (presenting the cross point deriver with a more difficult signal) sizable losses in SNR are experienced. On the other hand, as expected the third and fifth order algorithms yield the best results with nearly identical SNR figures that vary linearly with modulation depth. Again we see that the performance of the two sample consecutive UPWM DAC lies between that of the first order and the third order PNPWM systems. It is interesting to note that the SNR of the two sample consecutive system actually decreases as the amplitude of the input signal is increased to near full scale. This is because the increase in modulation depth results in the emergence of a third harmonic which for lower M was buried beneath the noise floor.

7.4.2 Discussion of Errors in Cross Point Derivation Algorithms

We now examine the structure of the errors associated with each cross point derivation procedure. Recall from Chapter Five that the main sources of error in the algorithms are (a) errors inherent to the rootfinding procedure, (b) signal/derivative approximation errors due to the low order of the interpolation polynomial, and (c) signal/derivative approximation errors due to the finite wordlength of the data used to form the interpolation polynomial. It is instructive to isolate these effects to determine which has the greatest influence on the total error.

We begin by comparing the output spectra of the perfect, first order, third order, and fifth order cross point derivers when driven by a 24 bit M=0.8, f_v=20.001kHz, $f_s(=f_c)$=352.8kHz tone input. These are shown in Figs. 7.38a-d. We see that the perfect cross point deriver's output consists of a 20.001kHz tone and harmonics of decreasing size. We also see three high frequency tones which are not harmonics of the input but are spaced about 20kHz apart. These are recognized as corresponding to foldback distortion terms for m=1 and n=–9,–10, and –11. The output of the first order cross point deriver also possesses harmonics of the input tone at levels similar to those of the perfect output. The foldback terms are absent. The third order output has components at the input harmonic frequencies as well as the foldback term frequencies. There are also several spurious components about many of the forward harmonics. The fifth order output is similar aside from the fact that the spurious terms seen in the third order case are smaller. As an interesting comparison Fig. 7.38e shows the output of a cross point deriver which implements a version of the Newton-Raphson procedure used in the third and fifth order cross point algorithms. However, it is similar to the perfect cross point deriver in

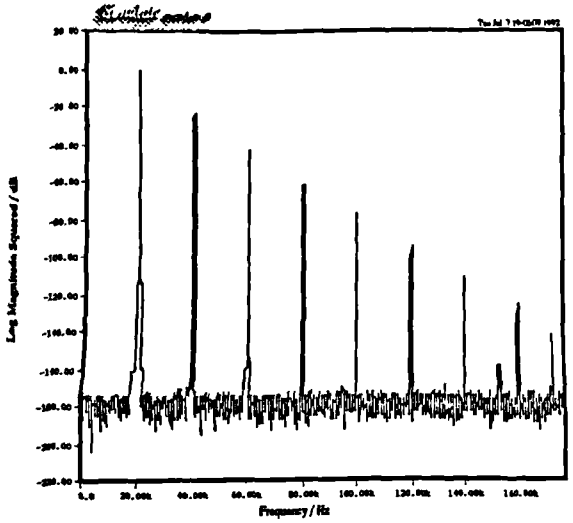


Fig. 7.38a: Cross Point Deriver Output Spectrum (PNPWM f)

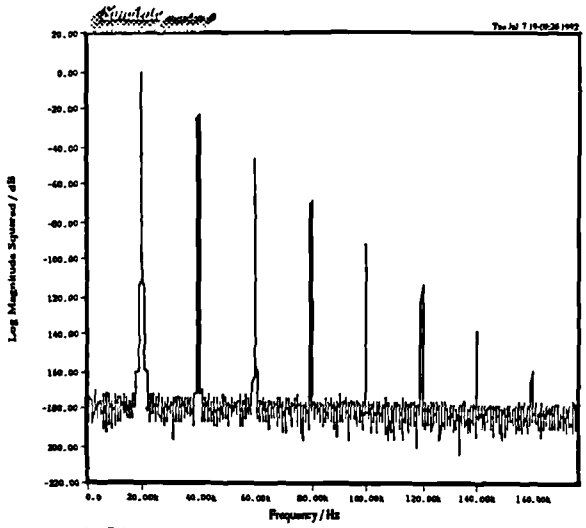


Fig. 7.38b: Cross Point Deriver Output Spectrum (PNPWM h)

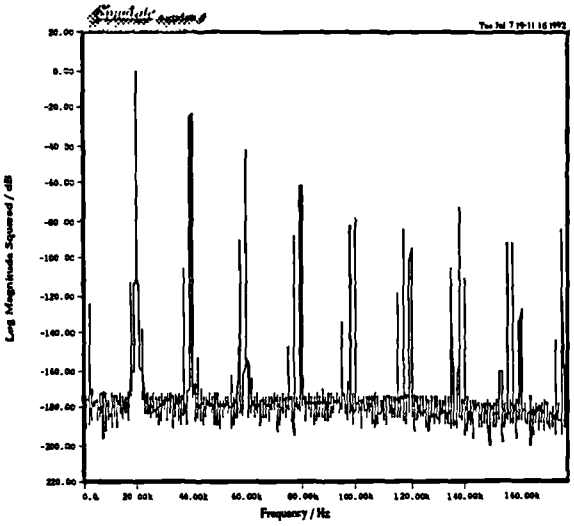


Fig. 7.38c: Cross Point Deriver Output Spectrum (PNPWM i)

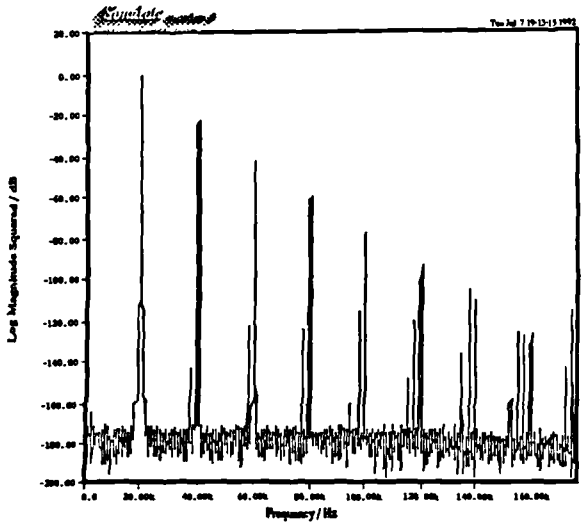


Fig. 7.38d: Cross Point Deriver Output Spectrum (PNPWM j)

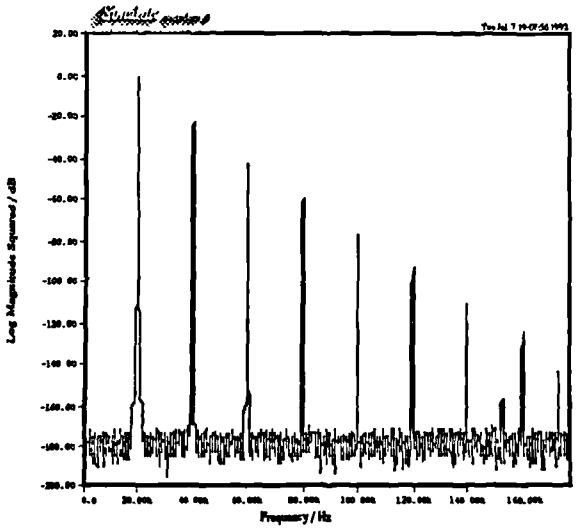


Fig. 7.38e: Cross Point Deriver Output Spectrum (PNPWM g)

that it uses direct calls to the sine and cosine functions—eliminating the use of polynomial approximation to the signal and its derivative. It therefore represents a means by which we can evaluate the underlying technique independent of any polynomial approximation errors. We see from the figure that this "perfect Newton-Raphson" procedure produces an output whose spectrum appears to closely resemble that of the perfect cross point deriver's.

It is also interesting to examine the spectra of the difference between the output of each cross point algorithm with that of the perfect cross point deriver as shown in Fig. 7.39. This will allow us to get a feel for the type of errors each cross point deriver produces. The error spectra are shown in Fig. 7.40a-d for the perfect Newton Raphson procedure as well the fifth, third, and first order algorithms, respectively. Here we see that in all four cases the error manifests itself in a highly structured form. The plots (which are scaled relative to the level of a maximum amplitude tone) indicate the presence of the spurious tones generated by the third order and fifth order procedures. In all cases there are differing amounts of amplitude and/or phase errors on the harmonics of the input as well as on the foldback terms. Specifically, we see that the errors associated with the perfect Newton Raphson technique are quite small. As expected, though, the errors tend to grow in size as the order of the interpolation polynomial is decreased.

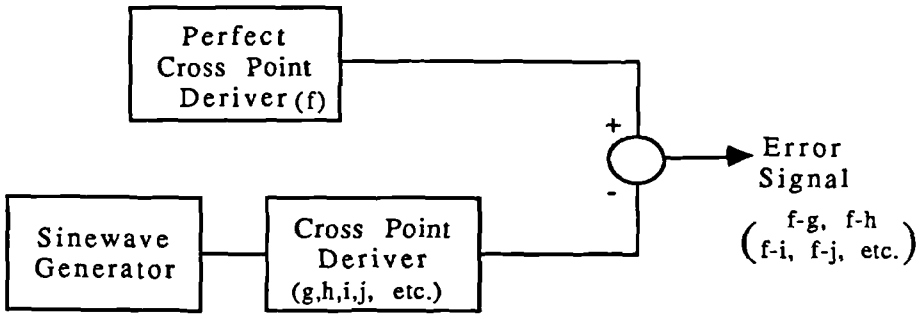


Fig. 7.39: Generation of Cross Point Deriver Error Signal

These tests are repeated for the 16 bit case. The error spectra are shown in Fig. 7.41a-d. Compared to the 24 bit results we see that the increased quantization noise levels in these plots mask many of the smaller tone errors we saw before. However, more significantly, it is clear that in the third and fifth order cases the greater noise on the input samples also has the effect of increasing the size of some of the error tones. Thus it is

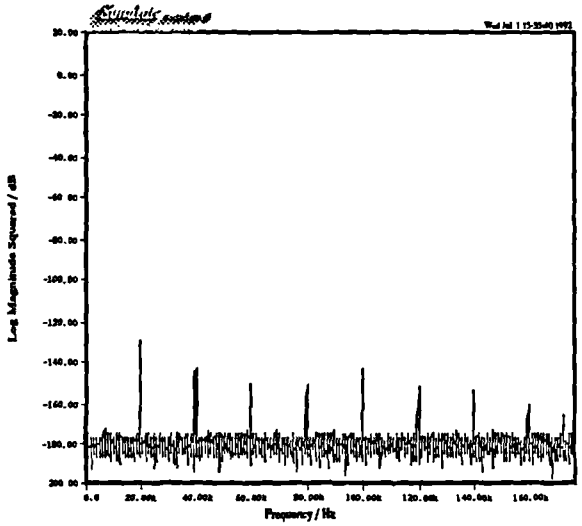


Fig. 7.40a: Cross Point Deriver Error Spectrum (24 bit PNPWM f-g)

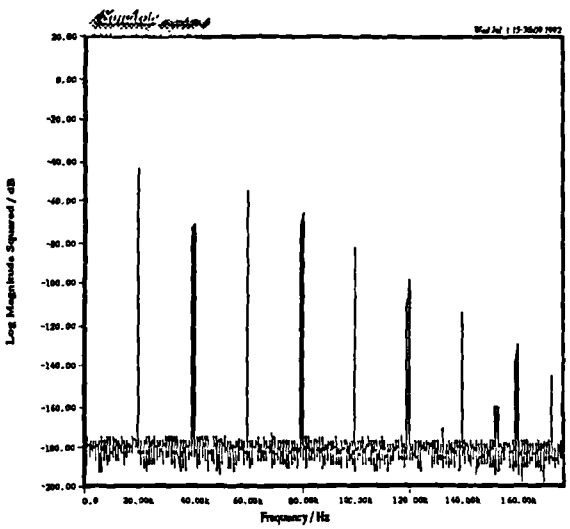


Fig. 7.40b: Cross Point Deriver Error Spectrum (24 bit PNPWM f-h)

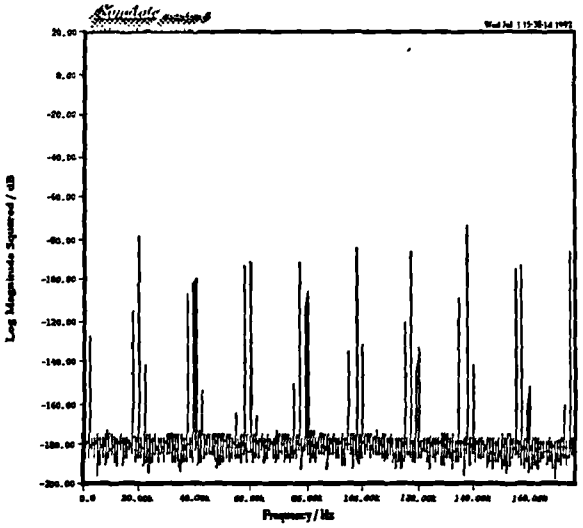


Fig. 7.40c: Cross Point Deriver Error Spectrum (24 bit PNPWM f-i)

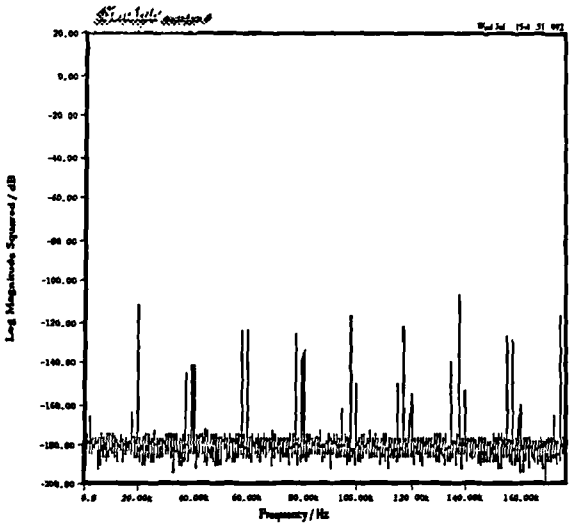


Fig. 7.40d: Cross Point Deriver Error Spectrum (24 bit PNPWM f-j)

incorrect to regard the presence of quantization noise on the input signal as simply resulting in independent, additive noise at the output.

We repeat these tests for the lower frequency, 1.001kHz input. The 24 bit and 16 bit error spectra are shown in Figs. 7.42a-d and 7.43a-d, respectively. For such a low frequency input the tone errors are confined to the baseband. Thus for clarity only the baseband region is shown. In general we see that these tone errors are smaller than those encountered for the 20.001kHz input. As before the errors are worse in the first order case. However, the other spectra are very similar which implies that the errors in the third and fifth order cases are due to the Newton Raphson technique itself rather than any other source of error. Also, as in the 20.001kHz case, the 16 bit difference spectra exhibit higher noise floors than in the 24 bit spectra.

Next, we present a more quantitative comparison of the cross point deriver difference signals. Specifically, in Tables 7.11a-c we compare the maximum value of the absolute value of the difference signal (i.e., the maximum difference between the perfect cross point deriver output and the output generated by each of the other procedures) as well as its mean squared power (i.e., the sum of the squares of the difference signal divided by the number of samples considered). We use input signals with M=0.8 and f_v=1.001kHz, 6.001kHz, and 20.001kHz and wordlengths of b=16, 20, and 24 bits. (In all cases the signals are scaled to be in the range of ±32768.) Generally speaking, the tables indicate that the high frequency, low wordlength, low polynomial order cases yield the worst results (i.e., the biggest maximum difference as well as the largest mean squared error). The relative insensitivity of the first order results for a given f_v to changes in wordlength indicates that the approximation errors associated with such a low order polynomial are the overall dominant source of error in these cases. On the other hand, for the fifth order algorithm, the errors are seen to be a combination of the type of approximation errors affecting the first order results as well as input signal quantization noise propagation effects. This is expected from the error analysis of Appendices 5A, 5C, and 5D. (Also, with the appropriate scaling, the maximum difference for the 20.001kHz fifth order algorithm is not far from ~$2.84\text{x}10^{-5}$, the upper bound derived in Appendix 5C: $1.26 \cdot 2^{-16} \approx 1.92\text{x}10^{-5}$.) In addition, as expected, in all cases, the perfect Newton Raphson technique yields the best results.

Another interesting set of difference signal results are those obtained by versions of the third and fifth order cross point deriver algorithms which are designed to compute only a single cross point per polynomial (modulation types k and l of Table 7.1). In each case only the cross point time in the centremost sub-interval over the support abscissae is computed (e.g., $[\tau_0,\tau_1]$, see Chapter Five). The maximum difference and error power results

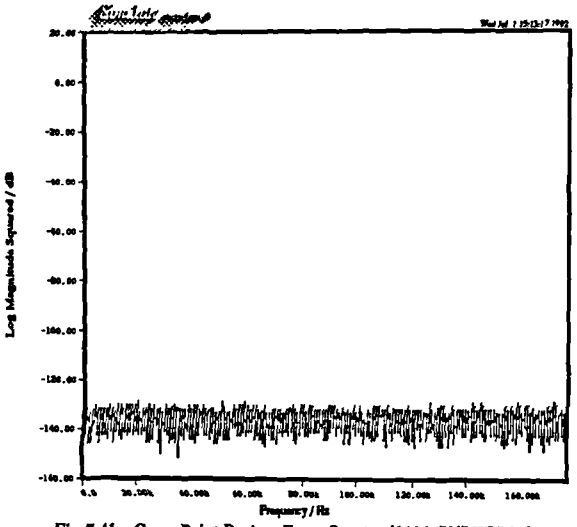


Fig. 7.41a: Cross Point Deriver Error Spectra (16 bit PNPWM f-g)

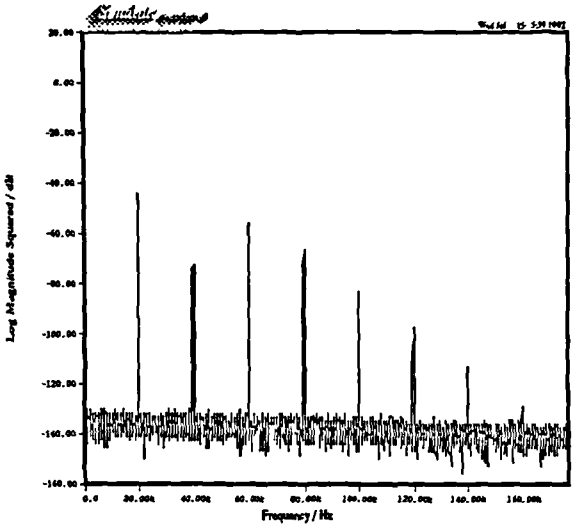


Fig. 7.41b: Cross Point Deriver Error Spectrum (16 bit PNPWM f-h)

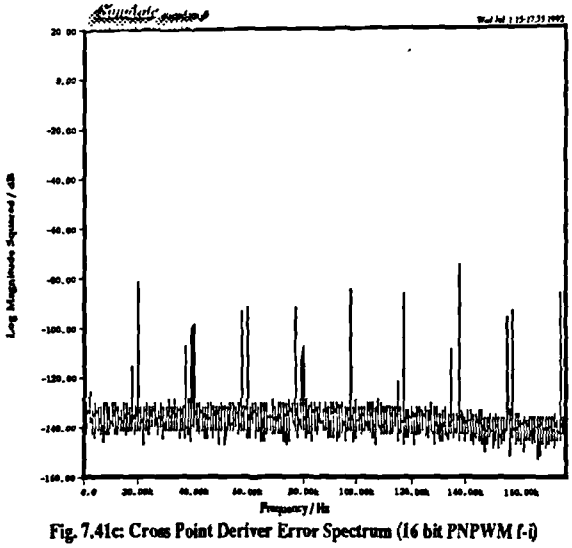


Fig. 7.41c: Cross Point Deriver Error Spectrum (16 bit PNPWM f-i)

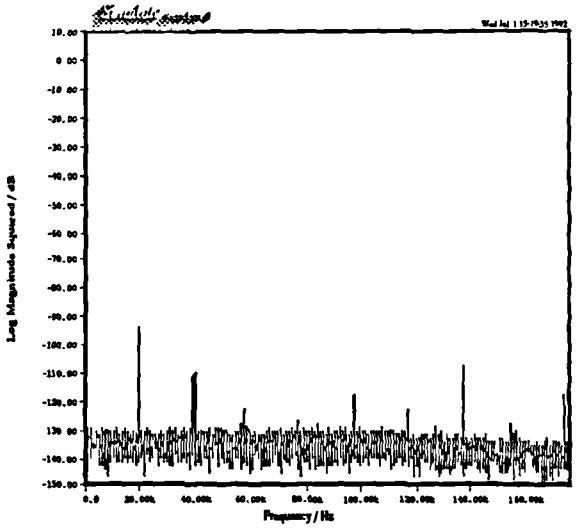


Fig. 7.41d: Cross Point Deriver Error Spectrum (16 bit PNPWM f-j)

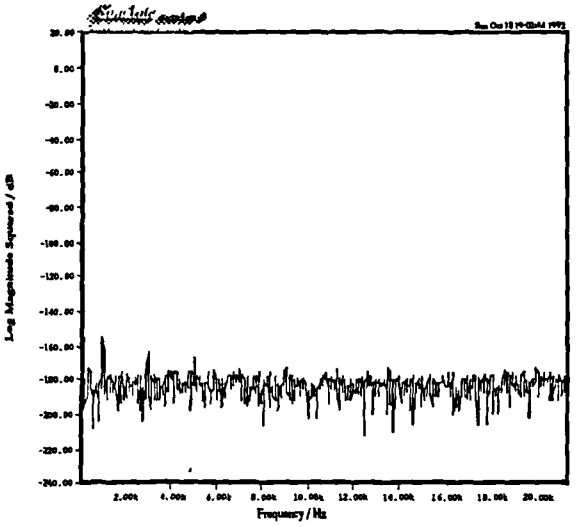


Fig. 7.42a: Cross Point Deriver Error Spectrum (24 bit PNPWM f-g)

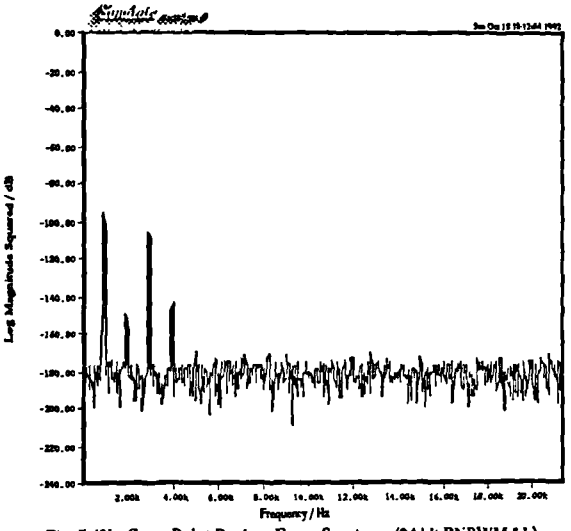


Fig. 7.42b: Cross Point Deriver Error Spectrum (24 bit PNPWM f-h)

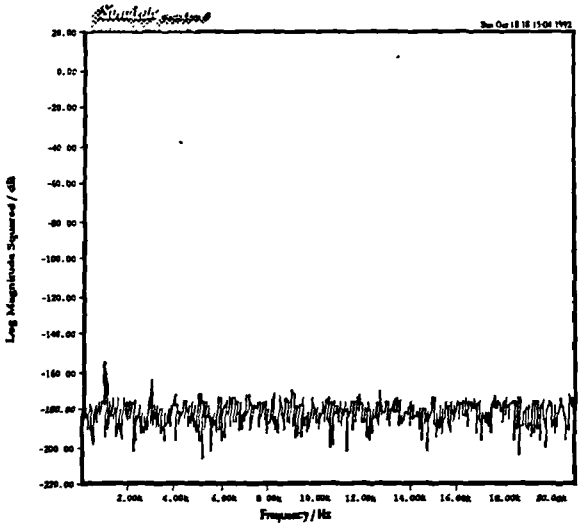


Fig. 7.42c: Cross Point Deriver Error Spectrum (24 bit PNPWM f-i)

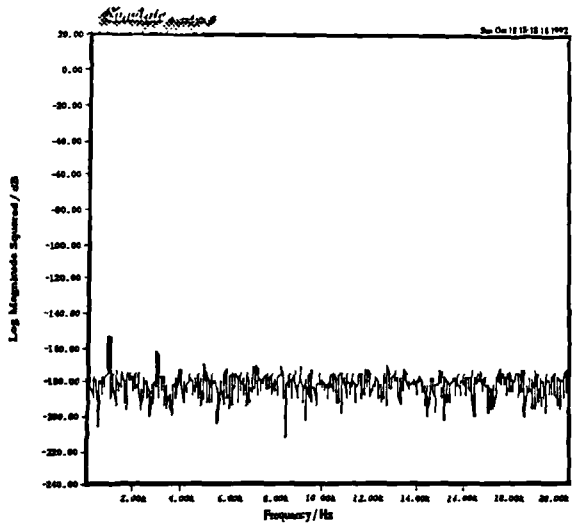


Fig. 7.42d: Cross Point Deriver Error Spectrum (24bit PNPWM f-j)

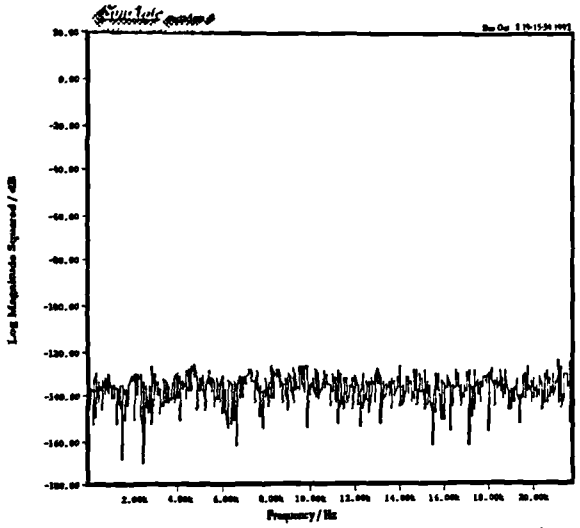


Fig. 7.43a: Cross Point Deriver Error Spectrum (16 bit PNPWM f-g)

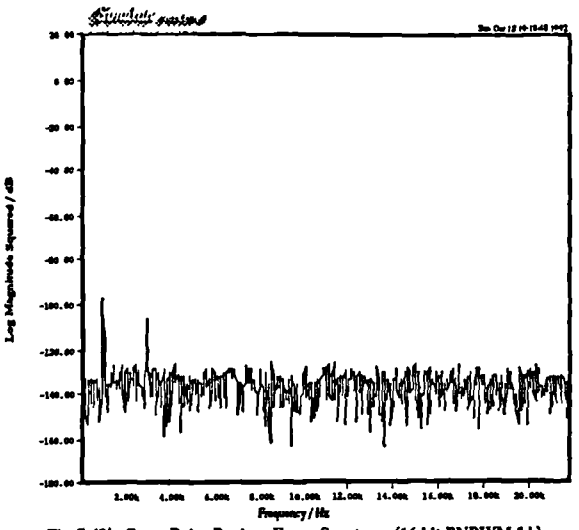


Fig. 7.43b: Cross Point Deriver Error Spectrum (16 bit PNPWM f-h)

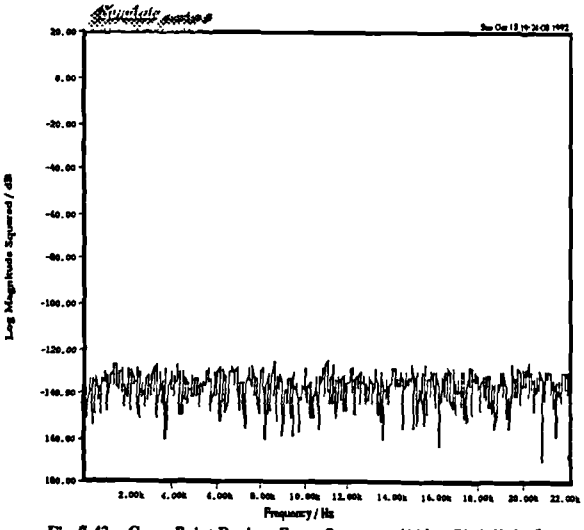


Fig. 7.43c: Cross Point Deriver Error Spectrum (16 biit PNPWM f-i)

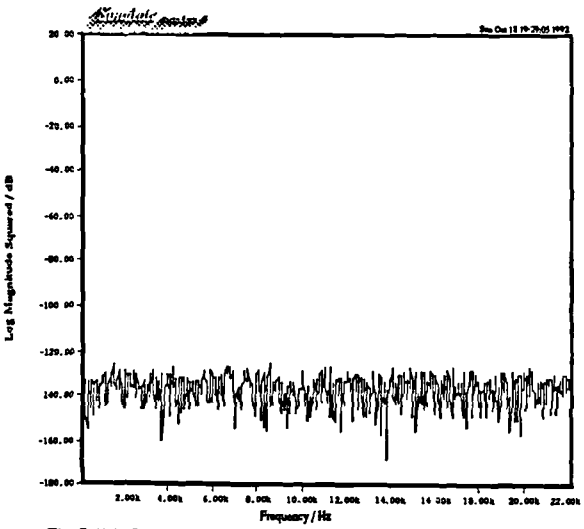


Fig. 7.43d: Cross Point Deriver Error Spectrum (16 bit PNPWM f-j)

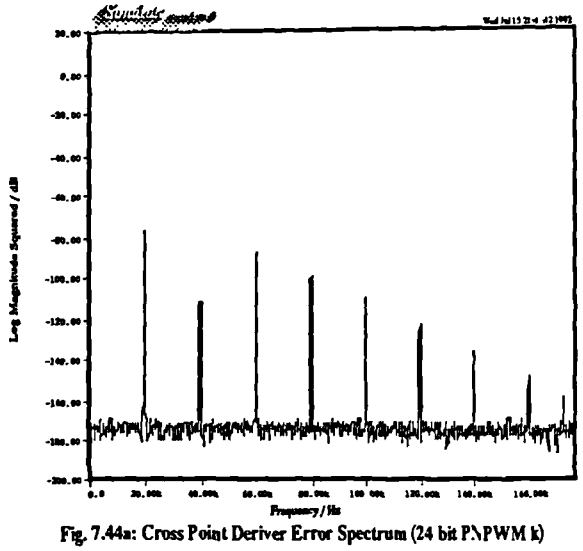


Fig. 7.44a: Cross Point Deriver Error Spectrum (24 bit PNPWM k)

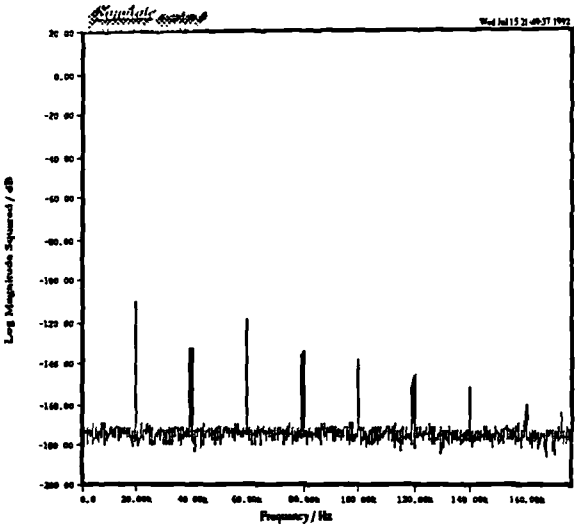


Fig. 7.44b: Cross Point Deriver Error Spectrum (24 bit PNPWM l)

Table 7.11a: Cross Point Deriver Errors for 1.001kHz input								
Modulation Type	*f*–*g*		*f*–*h*		*f*–*i*		*f*–*j*	
	max diff	mse	max diff	mse	max dif	mse	max dif	mse
b=16	$6.25x10^{-1}$	$9.95x10^{-2}$	$1.57x10^{0}$	$4.11x10^{-1}$	$1.23x10^{0}$	$1.77x10^{-1}$	$1.25x10^{0}$	$1.74x10^{-1}$
b=20	$3.91x10^{-2}$	$3.95x10^{-4}$	$5.65x10^{-1}$	$1.67x10^{-1}$	$8.11x10^{-2}$	$6.92x10^{-4}$	$7.60x10^{-2}$	$6.86x10^{-4}$
b=24	$2.44x10^{-3}$	$1.49x10^{-6}$	$5.06x10^{-1}$	$1.61x10^{-1}$	$4.79x10^{-3}$	$2.62x10^{-6}$	$4.57x10^{-3}$	$2.61x10^{-6}$

Table 7.11b: Cross Point Deriver Errors for 6.001kHz input								
Modulation Type	*f*–*g*		*f*–*h*		*f*–*i*		*f*–*j*	
	max diff	mse	max diff	mse	max dif	mse	max dif	mse
b=16	$6.30x10^{-1}$	$1.00x10^{-1}$	$1.92x10^{1}$	$2.11x10^{2}$	$1.27x10^{0}$	$1.72x10^{-1}$	$1.27x10^{0}$	$1.74x10^{-1}$
b=20	$3.94x10^{-2}$	$3.89x10^{-4}$	$1.83x10^{1}$	$2.08x10^{2}$	$1.45x10^{-1}$	$2.65x10^{-3}$	$7.55x10^{-2}$	$6.90x10^{-4}$
b=24	$2.45x10^{-3}$	$1.49x10^{-6}$	$1.82x10^{1}$	$2.08x10^{2}$	$9.43x10^{-2}$	$2.32x10^{-2}$	$4.66x10^{-3}$	$2.55x10^{-6}$

Table 7.11c: Cross Point Deriver Errors for 20.001kHz input								
Modulation Type	*f*–*g*		*f*–*h*		*f*–*i*		*f*–*j*	
	max diff	mse	max diff	mse	max dif	mse	max dif	mse
b=16	$.640x10^{-1}$	$9.68x10^{-2}$	$2.09x10^{2}$	$2.60x10^{4}$	$1.29x10^{1}$	$3.59x10^{1}$	$1.26x10^{0}$	$1.79x10^{-1}$
b=20	$4.00x10^{-2}$	$3.48x10^{-4}$	$2.08x10^{2}$	$2.59x10^{4}$	$1.22x10^{1}$	$3.64x10^{1}$	$2.90x10^{-1}$	$1.75x10^{-2}$
b=24	$1.56x10^{-2}$	$6.97x10^{-5}$	$2.08x10^{2}$	$2.59x10^{4}$	$1.22x10^{1}$	$3.65x10^{1}$	$2.52x10^{-1}$	$1.76x10^{-2}$

are shown in Tables 7.12a-c. On the whole, these figures indicate better performance than that associated with the original third and fifth order algorithms. This is to be expected since the polynomial approximation tends to be most accurate towards the centre of the interval spanned by the support abscissae. However, we note that since these procedures update the polynomial after each output sample they are more computationally intensive than the respective original algorithms.

Fig. 7.44a-b shows the output spectra of the error signals associated with these modified third and fifth order procedures for the 24 bit, 20.001kHz input. Here we see a complete absence of the spurious tones generated in the original third and fifth order algorithms. The errors appear to be concentrated solely on the input tone, its harmonics, and the foldback terms. Therefore, it is believed that the spurious tones associated with the original procedures are due to the strong correlation between polynomial approximation errors over the interval spanned by the support abscissae. (The approximation errors over the interval are not random but in fact form smooth, continuous, signal dependent error functions—one function for each polynomial.) When there is a new polynomial for each cross point there is less correlation between the errors on the signal and derivative approximations used to form the Newton Raphson iterate. Indeed, we see from Fig. 7.38b that even the poor quality first order procedure (which inherently uses a new polynomial for each cross point) does not exhibit these spurious tones.

It is also worthwhile to consider in more detail the precise cause of the errors in the output of some of the cross point derivers. We know that both the errors inherent to polynomial approximation as well as those due to quantization noise on the samples used to form the polynomial contribute to the overall error. However, we have not considered on an experimental basis whether this overall error is influenced more strongly by the manifestation of approximation errors in the signal approximation or in the derivative approximation. Therefore, we compare the performance of third and fifth order "hybrid" cross point derivers which are based on the Newton Raphson procedure but use either a "perfect" input signal (i.e., evaluated via calls to the sine function) or a perfect derivative. (These systems are used only for analysis and, of course, are not intended for use in a realistic DAC.) In this way we can isolate the effects of the polynomial approximation to the signal and to the derivative. Systems using these four hybrid cross point derivers are denoted by letters m through p in Table 7.1. Tables 7.13a-c indicate the performance. As could be expected, inspection of the results indicate that the performance of these systems lie somewhere in between that of the perfect Newton Raphson (g) and those of the ordinary third and fifth order algorithms (i and j, respectively). However, the tables also indicate that much larger errors are incurred when the polynomial approximations to the signal are used as opposed

Table 7.12a: One Cross Point per Polynomial Cross Point Deriver Errors (1.001kHz input)				
Modulation Type	$f-k$		$f-l$	
	max diff	mse	max diff	mse
b=16	$1.12\text{x}10^{0}$	$1.65\text{x}10^{-1}$	$1.17\text{x}10^{0}$	$1.69\text{x}10^{-1}$
b=20	$7.01\text{x}10^{-2}$	$6.46\text{x}10^{-4}$	$7.16\text{x}10^{-2}$	$6.60\text{x}10^{-4}$
b=24	$4.40\text{x}10^{-3}$	$2.48\text{x}10^{-6}$	$4.43\text{x}10^{-3}$	$2.53\text{x}10^{-6}$

Table 7.12b: One Cross Point per Polynomial Cross Point Deriver Errors (6.001kHz input)				
Modulation Type	$f-k$		$f-l$	
	max diff	mse	max diff	mse
b=16	$1.17\text{x}10^{0}$	$1.73\text{x}10^{-1}$	$1.19\text{x}10^{0}$	$1.68\text{x}10^{-1}$
b=20	$1.05\text{x}10^{-1}$	$1.98\text{x}10^{-3}$	$7.30\text{x}10^{-2}$	$6.68\text{x}10^{-4}$
b=24	$4.15\text{x}10^{-2}$	$9.00\text{x}10^{-4}$	$4.66\text{x}10^{-3}$	$2.55\text{x}10^{-6}$

Table 7.12c: One Cross Point per Polynomial Cross Point Deriver Errors (20.001kHz input)				
Modulation Type	$f-k$		$f-l$	
	max diff	mse	max diff	mse
b=16	$5.71\text{x}10^{0}$	$1.39\text{x}10^{1}$	$1.27\text{x}10^{0}$	$1.98\text{x}10^{-1}$
b=20	$4.67\text{x}10^{0}$	$1.29\text{x}10^{1}$	$1.79\text{x}10^{-1}$	$8.96\text{x}10^{-3}$
b=24	$4.62\text{x}10^{0}$	$1.29\text{x}10^{1}$	$1.20\text{x}10^{-1}$	$7.13\text{x}10^{-3}$

to polynomial approximations to its derivative. In fact, in the case with the perfect signal and the fifth order derivative approximation the results are very close to those encountered in the perfect Newton Raphson procedure. These results are consistent with the error analysis of Appendix 5C where it is seen that errors on the signal have a greater effect on the overall error than those on the derivative. All of this indicates that it may be desirable to put more effort into obtaining a higher quality approximation to the signal than to the derivative.

Lastly, to gain further insight into the overall *consequences* of these errors it is useful to examine the spectra of the difference between the input (actually, a very high resolution version of the input) and the final output of the DAC. We do so for the eight times times oversampling, eight bit, minimum phase NTF system with first order, third order, and fifth order cross point derivers. First we consider the M=0.8, f_v=20.001kHz case (essentially the worst case input for the DAC). The spectra of the error signals are shown in Figs. 7.45a-c. The large tone at the signal frequency for the first order case indicates the presence of amplitude and/or phase errors at the signal frequency introduced by the crude cross point algorithm. These errors become smaller as the order of the algorithm is increased. In all three cases we see a slight increase in high frequency baseband noise. This is due to effects shown earlier in Fig. 7.24b.

Results of similar tests are shown in Figs. 7.46a-c where M=0.8 and f_v=1.001kHz. Beginning with the first order algorithm we note the presence errors at the input frequency and at its third harmonic. However, compared to the 20.001kHz case, the errors are much smaller for this low frequency input. Moreover, there are no visible errors in the third order and fifth order cases.

7.4.3 Overview

In this section we have seen that the application of a cross point derivation algorithm prior to modulation can improve the linearity of ONS/PWM DACs. Specifically, performance of systems with first order, third order, and fifth order procedures has been investigated. All of these algorithms were found to perform best for low frequency, low amplitude inputs. The crude but computationally efficient first order procedure reduced levels of harmonic and intermodulation distortion while the computationally intensive fifth order procedure succeeded in eliminating most visible forms of distortion. The performance of the third order system was marginally worse than that of the fifth order algorithm but much better than that of the first order procedure.

Table 7.13a: More Cross Point Deriver Errors for 1.001kHz input								
Modulation Type	*f*–*m*		*f*–*n*		*f*–*o*		*f*–*p*	
	max diff	mse	max diff	mse	max dif	mse	max dif	mse
b=16	$1.23\text{x}10^{0}$	$1.77\text{x}10^{-1}$	$6.25\text{x}10^{-1}$	$9.95\text{x}10^{-2}$	$1.25\text{x}10^{0}$	$1.74\text{x}10^{-1}$	$6.25\text{x}10^{-1}$	$9.95\text{x}10^{-2}$
b=20	$8.11\text{x}10^{-2}$	$6.92\text{x}10^{-4}$	$3.91\text{x}10^{-2}$	$3.95\text{x}10^{-4}$	$7.60\text{x}10^{-2}$	$6.86\text{x}10^{-4}$	$3.91\text{x}10^{-2}$	$3.95\text{x}10^{-4}$
b=24	$4.79\text{x}10^{-3}$	$2.62\text{x}10^{-6}$	$2.44\text{x}10^{-3}$	$1.48\text{x}10^{-6}$	$4.57\text{x}10^{-3}$	$2.61\text{x}10^{-6}$	$2.44\text{x}10^{-3}$	$1.48\text{x}10^{-6}$

Table 7.13b: More Cross Point Deriver Errors for 6.001kHz input								
Modulation Type	*f*–*m*		*f*–*n*		*f*–*o*		*f*–*p*	
	max diff	mse	max diff	mse	max dif	mse	max dif	mse
b=16	$1.27\text{x}10^{0}$	$1.72\text{x}10^{-1}$	$6.30\text{x}10^{-1}$	$1.00\text{x}10^{-1}$	$1.27\text{x}10^{0}$	$1.74\text{x}10^{-1}$	$6.30\text{x}10^{-1}$	$1.00\text{x}10^{-1}$
b=20	$1.45\text{x}10^{-1}$	$2.66\text{x}10^{-3}$	$3.94\text{x}10^{-2}$	$3.90\text{x}10^{-4}$	$7.55\text{x}10^{-2}$	$6.90\text{x}10^{-4}$	$3.94\text{x}10^{-2}$	$3.89\text{x}10^{-4}$
b=24	$9.44\text{x}10^{-2}$	$2.32\text{x}10^{-3}$	$2.54\text{x}10^{-3}$	$1.52\text{x}10^{-6}$	$4.66\text{x}10^{-3}$	$2.55\text{x}10^{-6}$	$2.45\text{x}10^{-3}$	$1.49\text{x}10^{-6}$

Table 7.13c: More Cross Point Deriver Errors for 20.001kHz input								
Modulation Type	*f*–*m*		*f*–*n*		*f*–*o*		*f*–*p*	
	max diff	mse	max diff	mse	max dif	mse	max dif	mse
b=16	$1.30\text{x}10^{1}$	$3.67\text{x}10^{1}$	$7.59\text{x}10^{-1}$	$1.09\text{x}10^{-1}$	$1.26\text{x}10^{0}$	$1.80\text{x}10^{-1}$	$6.48\text{x}10^{-1}$	$9.70\text{x}10^{-2}$
b=20	$1.23\text{x}10^{1}$	$3.72\text{x}10^{1}$	$1.80\text{x}10^{-1}$	$5.51\text{x}10^{-3}$	$2.91\text{x}10^{-1}$	$1.78\text{x}10^{-2}$	$5.39\text{x}10^{-2}$	$4.13\text{x}10^{-4}$
b=24	$1.23\text{x}10^{1}$	$3.73\text{x}10^{1}$	$1.43\text{x}10^{-1}$	$4.80\text{x}10^{-3}$	$2.54\text{x}10^{-1}$	$1.79\text{x}10^{-2}$	$3.19\text{x}10^{-2}$	$1.33\text{x}10^{-4}$

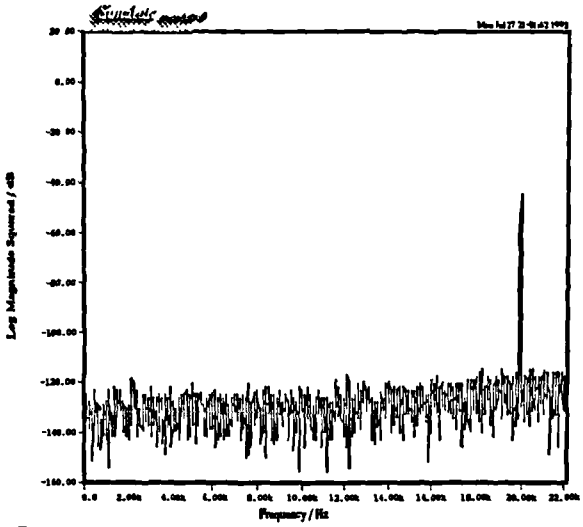


Fig. 7.45a: Output Error Spectrum for 20kHz Input (8x MP NTF PNPWM h)

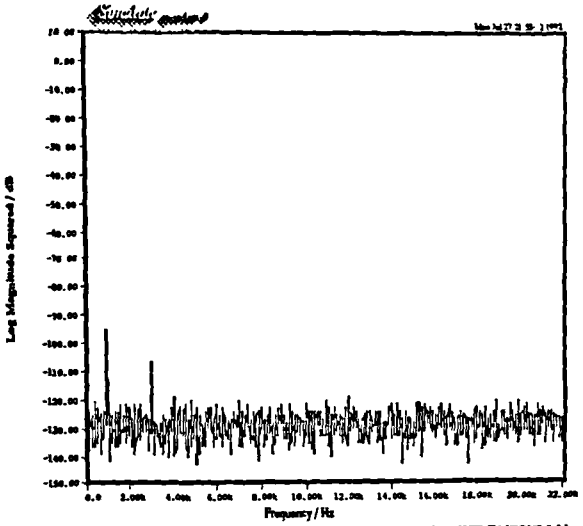


Fig. 7.46a: Output Error Spectrum for 1kHz Input (8X MP NTF PNPWM h)

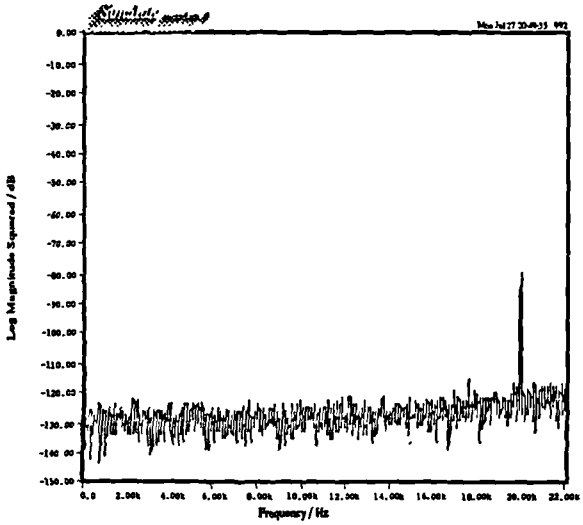


Fig. 7.45b: Output Error Spectrum for 20kHz Input (8X MP NTF PNPWM i)

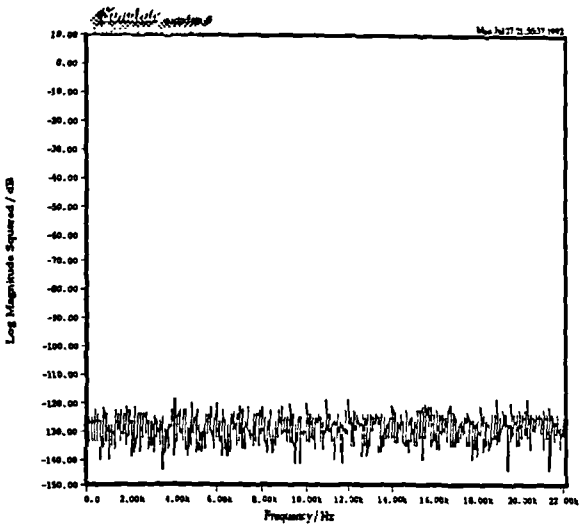


Fig. 7.46b: Output Error Spectrum for 1kHz Input (8X MP NTF PNPWM i)

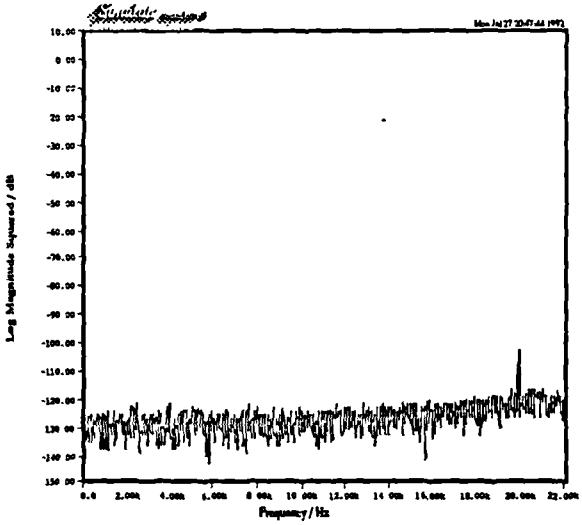


Fig. 7.45c: Output Error Spectrum for 20kHz Input (8X MP NTF PNPWM j)

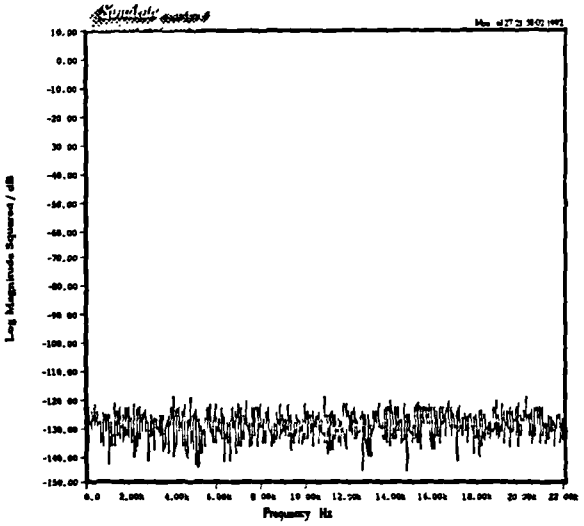


Fig. 7.46c: Output Error Spectrum for 1 kHz Input (8X MP NTF PNPWM j)

Errors due to the cross point deriver were shown to be a function of the basic rootfinding procedure itself, the low order of the polynomial approximation to the signal and its derivative, as well as quantization noise effects. It was also seen that in general these types of algorithms are much more sensitive to errors in the approximation of the signal as opposed to the derivative. Errors in the output of the DAC with the first order cross point deriver were found to be highly signal dependent, taking the form of harmonic distortion and errors at the input frequency. (As mentioned before, there is some preliminary evidence to suggest that the latter types of errors are not as severe as some of the extremely low SNR figures presented earlier would suggest.) The errors in the third order algorithm were found to be less signal dependent. The errors in the fifth order procedure were noise-like and nearly independent of the input signal. Interestingly, the two sample consecutive UPWM system was found to perform better than the first order PNPWM but considerably worse than the third and particularly the fifth order PNPWM systems.

7.5 Summary

In this chapter we have examined in detail the performance of many PWM based DAC systems. We began by considering the basic UPWM DAC structure of Fig. 7.1 with five different UPWM modulation types. It was shown that in each case by raising the pulse repetition frequency of the modulator (with a corresponding increase in the sampling rate of the digital signal) UPWM harmonic distortion, foldback distortion, and intermodulation distortion all could be reduced.

While oversampling helped to improve the linearity of UPWM DACs, it further exacerbated the severe problem of excessive modulator clock speed. For a 16 bit audio system even without oversampling, the modulator would be required to operate at rates in the GigaHertz. To solve this problem oversampled noise shaping was used to reduce the modulator clock speed by reducing the wordlength of the oversampled signal applied to the modulator with negligible loss in SNR. Hence with ONS/UPWM DACs of the form shown in Fig. 7.9 higher levels of performance could be obtained at reasonable modulator clock speeds. However, certain undesirable effects arising from the combination of noise shaping and PWM were seen to degrade the performance of some systems. To combat this it was shown that the careful selection of noise shaper NTF could, in some cases, substantially reduce these effects. In general, for a fixed pulse repetition frequency it was found that two sample consecutive UPWM offered the best performance.

Unfortunately, irrespective of UPWM modulation type, true 16 bit quality performance could not be obtained at reasonable pulse repetition frequencies. For this reason, we introduced the idea of using a premodulation signal processing algorithm designed to linearize the baseband performance of conventional trailing edge UPWM modulators. Estimates to the natural sampling cross point times are obtained digitally and applied to the conventional modulator. These so-called ONS/PNPWM DACs, shown in Fig. 7.29, effectively implement a completely digital version of "distortion free" NPWM. The performance of systems with three cross point algorithms of varying complexity was investigated. It was shown that virtually distortion free performance was possible but at relatively high levels of computational complexity. The question of which specific DAC is the most appropriate for implementation is addressed in the next chapter.

Chapter Eight
Conclusions

This thesis has presented a class of high resolution DACs which may be suitable for conventional low power use as well as high power applications. In this chapter we present an overview of the past seven chapters and indicate specific topics for investigation in future work.

8.1 Summary

Chapter One set the context for the work described in this thesis. It can be thought of as a continuation of that carried out by Sandler [Sa83] in the context of digital power amplification (i.e., the direct conversion of a digital signal into the corresponding analogue power waveform with no intermediate low power DAC stage). We may also view the work in the general context of low power oversampling DACs.

In Chapter Two the various PWM modulation types considered in this thesis were introduced. In a continuous time analogue sense, these modulation types are broadly distinguished by the method in which samples modulating the PWM waveform are chosen. With uniform sampling PWM (UPWM) the sampling instants of the samples used to modulate the PWM waveform are uniformly spaced in time. By contrast, in natural sampling PWM (NPWM) the sampling instants are nonuniformly spaced and signal dependent. In both UPWM and NPWM it is possible to modulate a single edge or both edges of the pulse. Double sided NPWM uses distinct samples to modulate the leading and trailing edges of the pulse. However, in the UPWM case it is possible to use two samples per pulse or just a single sample, giving rise to two sample consecutive UPWM and symmetric double sided UPWM, respectively. Also, when considering digital UPWM specifically, various types of asymmetric modulation can be produced.

Where possible expressions for the tone spectra of the various modulation types were presented. In general, it was found that PWM tone spectra are comprised of tones at the

input signal frequency, its harmonics, the pulse repetition frequency (the carrier), its harmonics, and sideband terms spaced at multiples of the input frequency about the carrier and its harmonics. Important exceptions to this include the absence of signal harmonic components (i.e., harmonic distortion) in both single sided and double sided NPWM as well as the absence of even order signal harmonics in two sample consecutive UPWM.

In terms of tone modulation baseband performance, particular concern must be given to the levels of harmonic distortion (second, third, and sometimes, fourth order) as well as so-called foldback distortion, which is comprised of sideband components (particularly those about the carrier itself) that fall into the baseband. For each modulation type simple approximations to the levels of such components were derived.

While in NPWM there is no harmonic distortion, such components do exist in the UPWM spectra. However, NPWM foldback distortion is often larger than that of the corresponding UPWM modulation type. Moreover, in general, the double sided modulation types were found to exhibit lower levels of distortion than the corresponding single sided modulation types.

In addition, it was shown that increasing the pulse repetition frequency of the PWM waveform tended to reduce all forms of baseband distortion. In terms of our 16 bit quality, 20kHz baseband audio application, it was found that in all cases baseband foldback distortion could be reduced to negligible levels with relatively mild increases in pulse repetition frequency. Unfortunately, however, *excessive* increases in pulse repetition frequency are required to reduce UPWM harmonic distortion to negligible levels.

Chapter Three was devoted to sample rate conversion. Interpolation and decimation, sample rate increase and sample rate decrease, respectively, are important in the hardware implementation as well as the software simulation of PWM based DACs. In particular, an interpolator is required to raise the sampling rate of the DAC input signal to a rate commensurate with modulator's pulse repetition frequency (which is often increased above the Nyquist minimum to reduce PWM distortion). A decimator is needed in order to facilitate baseband analysis of computer simulations of PWM based DACs.

We began the chapter with a review of the basic techniques used to alter the sampling rate of a signal. Interpolation and decimation were seen to be essentially complementary digital filtering procedures. Techniques for reducing the computational complexity of sample rate conversion tasks were also discussed. These included the removal of unnecessary computation in the basic sample rate conversion methods, the decomposition of a sample rate converter into a cascade of smaller, independent conversion stages, and the use of one of several classes of filters which enable further reductions in computation.

Chapter Four focused on oversampled noise shaping (ONS) networks. One of the most serious problems faced in the implementation of a 20kHz bandwidth, 16 bit quality PWM DAC is that of excessive modulator clock speed (which is in the GigaHertz—even without oversampling). The clock rate is exponentially related to the input signal wordlength. As such, modulator clock speed can be reduced substantially by lowering the wordlength of the input. However, this generally results in a loss of signal quality. ONS techniques enable a reduction in the wordlength of an oversampled signal without appreciable loss in baseband SNR. Thus, ONS networks are useful for making a PWM based DAC more practical to implement.

Noise shaping was shown to rely on a combination of oversampling, coarse requantization, and error feedback. A simple linearized analysis allowed us to express the output of a noise shaper as a sum of the input and a spectrally shaped version of the requantization error. The requantization error is attenuated over the baseband at the expense of increased high frequency noise. On the basis of the analysis, we can accurately design ONS networks for a given SNR requirement. In particular, it was shown that 16 bit quality noise shapers can be designed to reduce the signal wordlength to the point where the corresponding PWM modulator clock speeds are low enough to be realized in hardware.

Other issues related to the tailoring of ONS networks for use with PWM were also addressed. Specifically, the theoretical basis of a class of low noise power gain networks was presented. Also, networks with output error sequences possessing special correlation properties were described. Later in the thesis, use of these special noise shapers was shown to eliminate certain undesirable baseband noise effects.

Chapter Five explored techniques for improving DAC performance. In spite of NPWM being known to give better performance, UPWM had always formed the basis of PWM DACs. Given that digital signals often arise from (or at least can be thought of as arising from) uniform sampling of an analogue waveform, UPWM was viewed as the only practical option. (NPWM was believed to be suitable only for continuous time analogue inputs.) We sought to overcome this limitation by proposing a technique called Pseudo-Natural PWM (PNPWM). This approach allows us to achieve excellent, NPWM performance but in a fully digital implementation. The idea is to digitally approximate the NPWM sampling instant and to apply this approximation to a conventional UPWM modulator. In practice this can be done by forming a polynomial approximation to the underlying analogue signal (quantized samples of which form the digital input signal) and applying a rootfinding procedure (the secant method and possibly the Newton-Raphson method) to numerically estimate the time of intersection between the analogue signal and the PWM comparison waveform. Specific algorithms of varying degrees of complexity and accuracy

were described in detail. Error analyses were performed as well.

In Chapter Six the structure of the software written to simulate the DACs considered in this thesis was presented. Although, the four previous chapters contained some theoretical characterization of the main PWM DAC sub-systems, it was believed that software simulations of the complete DACs were essential for gaining insight into overall performance levels. In particular, we would like to understand how some of the highly nonlinear sub-systems interact to effect DAC performance. Such knowledge is difficult to obtain by a purely analytical approach.

The simulation is structured as a collection of small, independent modules each of which mimics a stage within the DAC. The flexibility of the software enables relatively quick and simple analysis of a large variety of DAC configurations.

Chapter Seven presented results from computer simulations of several PWM DACs. We began with the basic oversampling UPWM DACs using a variety of modulation types. In particular, for tone modulation, the levels of harmonic distortion and foldback distortion were examined as a function of signal amplitude and pulse repetition frequency. Twin tone tests indicated the presence of intermodulation distortion as well. In general, it was shown that all forms of baseband distortion decreased as the pulse repetition frequency of the modulator was increased. Simulation results were found to correspond closely to theoretical predictions. Specifically, it was seen that single sided modulation appeared to exhibit the highest levels of baseband distortion followed by the double sided modulation types, with the two sample consecutive modulation frequently offering the lowest levels of total harmonic distortion.

Then noise shaping was used in conjunction with the systems described above. With the pulse repetition frequency fixed, the performance of each DAC was examined for a variety of noise shapers with different output wordlengths. For the DACs with mild noise shaping (i.e., small reductions in signal wordlength) the results strongly resembled those of the corresponding oversampling UPWM DACs (which did not use noise shaping). However, when the noise shapers with low output wordlength were used, undesirable effects were often observed in form of increased baseband output noise power. Qualitative explanations for these effects were offered. While more research is needed, the PWM process is believed to be such that high frequency noise shaper noise is modulated back into the baseband analogous to baseband foldback or intermodulation distortion for tone inputs. This is thought to be especially true for trailing edge UPWM. For two sample consecutive UPWM there were additional problems. These arise from the structure of the noise shaper output error sequence. In both cases specially designed ONS networks were successfully used to eliminate any visible anomalies in the output noise floor. Also, for asymmetric

double sided UPWM, it was found that certain undesirable effects could be lessened by alternating the asymmetry between the leading and trailing pulse edges. In general, the systems using noise shapers with low (eight bit) output wordlengths operated at rates low enough to be realized in hardware.

Lastly, the performance of ONS/PNPWM DACs was evaluated. Converters with the first, third, and fifth order cross point algorithms of Chapter Five were considered. These DACs were seen to be successful in reducing the levels of harmonic as well as intermodulation distortion associated with the corresponding conventional ONS/UPWM DACs. The computationally intensive fifth order algorithm yielded very high levels of performance—completely eliminating harmonic and intermodulation distortion. The efficient first order procedure did reduce distortion levels, but significant errors remained. The third order procedure was in between (but with performance closer to that of the fifth order procedure). SNR measurements were performed as a function of input frequency and amplitude. In the fifth order case, SNR was nearly constant with frequency and linearly related to amplitude. However, in the first and third order cases, SNR decreased as the signal frequency was increased. Also, for these two cases, overall DAC errors were found to be highly structured and highly signal dependent—particularly in the first order case. For the fifth order case, however, errors were less structured and less signal dependent.

For very high frequency inputs the primary source of error was in the magnitude and/or phase of the tone itself. (As such, these may not be as subjectively disastrous as the poor SNR figures may lead us to believe.) Attempts were made to examine the various sources of error within the cross point time algorithms themselves. In particular, tests were performed to isolate and examine the effects of the Newton Raphson procedure itself, the approximation errors due to the low order of the polynomial, and the noise propagation effects arising from quantization errors on the samples forming the polynomial. Additional tests indicated that the procedure was much more sensitive to errors in the approximation of the *signal* as opposed to that of the *signal derivative*.

8.2 Future Work

While this thesis has laid the basic groundwork for a new class of high resolution DACs, important practical and theoretical work has yet to be done. In this section we briefly describe topics for future research.

One difficult problem to be solved is that of properly characterizing the PWM modulation process for random noise inputs as well as for deterministic signals plus random

noise. In particular, we would like to know how the modulator modifies the power spectral density of a stationary stochastic input (i.e., how the modulator redistributes the input noise power over frequency). In this way we could determine which frequency bands give rise to increased baseband noise power in the output signal. This would allow us to design ONS NTFs in the knowledge that noise at the modulator's input will be concentrated in less sensitive frequency regions such that the undesirable baseband effects encountered in Section 5.3 of Chapter Five will be minimized. It may also provide guidelines on how to design NTFs which allow us to noise shape more aggressively (i.e., to considerably fewer than eight bits while retaining 16 bit quality in the baseband after modulation). This will permit further reductions in modulator clock speed.

Derivation of the PWM spectra for single tone inputs is long and tedious. In the stochastic case, a general expression for the power spectral density of the output in terms of the statistics of the input signal would probably be even more difficult. Nevertheless, some results do exist [Mi60, Ro65] and could form a basis on which to proceed. Whether or not a useful theoretical characterization can be obtained, a thorough experimental (i.e., computer simulation) investigation would certainly be valuable.

Also, generally speaking, NTF design should be expanded to include IIR feedback filters which can aid in improving the overall computational efficiency of the noise shaper.

In terms of the baseband linearization of PWM a more complete statistical analysis of the signal approximation/root-finding techniques would be useful. Also, further research into improving the computational efficiency of the linearization procedure is needed. This may take the form of investigating the other techniques mentioned in Section 5.4 of Chapter Five.

In addition, important new techniques for the baseband linearization of PWM have recently been reported. The first, in principle, is a generalization of the techniques used in this thesis and is based on a type of UPWM/NPWM hybrid [Me91]. The second uses a model based approach to linearizing PWM [Ha92]. Both strategies are capable of yielding impressive results. Full comparisons of the computational complexity and performance of these techniques and the algorithms presented in this thesis are necessary.

For digital amplification applications, important practical problems associated with the power switching stage (such as those described in Section of Chapter Two) must be investigated. This would probably take the form of constructing in hardware and evaluating the performance of several power switching circuits. Results of such an investigation may place tighter restrictions on pulse repetition frequency, modulator input wordlength, and

modulator clock speed than those observed in this thesis.*

Also, the use of feedback may prove to be valuable in the correction of nonlinearities in the power switching stage. Moreover, feedback may possibly serve as an alternative or complementary approach to the linearization of PWM generally. This is presently under investigation [Hi92d].

In addition, the linearization of only single sided PWM systems was addressed in this thesis. The linearization procedures can easily be adapted for double sided modulation. However, a big disadvantage of double sided systems is the need to compute *two* cross point times per pulse. Nevertheless, it may be possible to further lower the pulse repetition frequency distortion and to use lower order linearization procedures in the double sided case (where there is less foldback distortion and less harmonic distortion). A study is currently underway [Pa92a].

Lastly, while in Section 7.4 of Chapter Seven we have compared four DACs (three ONS/PNPWM DACs using first, third and fifth order cross point algorithms and one ONS/UPWM two sample consecutive DAC) little has been said on which specific procedure should be chosen. In terms of the PNPWM DACs we have seen that, as could be expected, there exists an inverse relationship between computational efficiency and accuracy. Also, while the two sample consecutive DAC gives performance slightly better than the first order PNPWM DAC, it must operate at twice the modulator clock speed (unless a more complicated NTF is used to noise shape down to seven bits). For audio applications subjective preference must also considered. It is therefore believed that a proper tradeoff between accuracy and efficiency can only be obtained with listening tests. For instance, there would be no purpose in considering the computationally intensive fifth order algorithm if its performance was subjectively indistinguishable from that of the third order procedure.

Work is currently underway to construct a full 16 bit quality ONS/PNPWM DAC prototype. In the long term, once a full 16 bit quaity hardware implementation has been

* Here we are referring particularly to limitations imposed by the power switches used in the amplifier and not to general limitations of the (low power) modulator itself. In terms of the latter, there is hardware evidence [Hi92d] indicating that the configuration used in the PNPWM DAC simulations (i.e., the eight bit modulator with pulse repetition frequency of 352.8kHz and modulator clock speed of ~90.3MHz) is acceptable. In any case, it is desirable generally to lower the pulse repetition frequency and to decrease clock speed. However, the lower pulse repetition frequency may make linearization more difficult (in spite of the longer time in which to carry out the computations). Also, the decreased modulator clock speed implies the need for more drastic noise shaping. This may result in the sort of baseband noise problems observed in Chapter Seven (unless, as mentioned earlier, progress is made in characterization of the modulation process for noise inputs such that special NTFs can be designed to avoid this problem).

realized, research on even higher resolution DACs may begin.

References

[Ar87] S.H. Ardalan and J.J. Paulos: "An Analysis of Nonlinear Behavior in Delta-Sigma Modulators," IEEE Transactions on Circuits and Systems, Vol. CAS-34, June 1987.

[At89] K.E. Atkinson: *An Introduction to Numerical Analysis*, Wiley, New York, 1989.

[Be33] W.R. Bennett: "New Results in the Calculation of Modulation Products," Bell Systems Technical Journal, Vol 12, 1933.

[Be48] W.R. Bennett: "Spectra of Quantized Signals," Bell Systems Technical Journal, Vol. 27, July 1948.

[Bl53] H.S. Black: *Modulation Theory*, Van Nostrand, 1953.

[Bl78] B.A. Blesser: "Digitization of Audio—A Comprehensive Examination of Theory, Implementation, and Current Practice," Journal of Audio Engineering Society, Vol. 26, No. 10, 1978.

[Bo75] S.R. Bowes and B.M. Bird: "Novel Approach to the Analysis and Synthesis of Modulation Processes in Power Converters," Proceedings of IEE, Vol. 122, No. 5, 1975.

[Bo92] R. Bowman: private discussion, 1992.

[Ca85] J.C. Candy: "A Use of Double Integration in Sigma Delta Modulation," IEEE Transactions on Communications, Vol. COM-33, No. 3, March 1985.

[Ca89] L.R. Carley: "A Noise-Shaping Coder Topology for 15+ Bit Converters," IEEE Journal of Solid-State Circuits, Vol. 24, No. 2, April 1989.

[Cr90] P.G. Craven: private communication, 1990.

[Cr83] L.E. Crochiere and L.R. Rabiner: *Multirate Digital Signal Processing*, Prentice-Hall, Inc., New Jersey, 1983.

[Cu60] C.C. Cutler: "Transmission Systems Employing Quantization," U.S. Patent No. 2927962, 1960.

[DS89] *DSP96002 IEEE Floating-Point Dual-Port Processor User's Manual*, Motorola, 1989.

[Ge78] A. Gersho: "Principles of Quantization," IEEE Transactions of Circuits and Systems, Vol. CAS-25, July 1978.

[Ge89] M. Gerzon and P.G. Craven: "Optimal Noise Shaping and Dither of Digital Signals," Presented at the 87th Convention of the Audio Engineering Society, New York, October 1989.

[Go89] J.M. Goldberg and M.B. Sandler: "The Application of Noise Shaping for an All Digital Audio Power Amplifier," Presented at the 87th Convention of the Audio Engineering Society, New York, October 1989.

[Go90a] J.M. Goldberg and M.B. Sandler: "New Results in PWM for Digital Power Amplification," Presented at the 89th Convention of the Audio Engineering Society, Los Angeles, September 1990.

[Go90b] J.M. Goldberg and M.B. Sandler: "Comparison of PWM Techniques for Digital Power Amplifiers," Proceedings of the Institute of Acoustics, Vol. 12, No. 10, 1990.

[Go91a] J.M. Goldberg and M.B. Sandler: "Noise Shaping and Pulse-Width Modulation for an All-Digital Audio Power Amplifier," Journal of the Audio Engineering Society, Vol. 39, No. 6, June 1991.

[Go91b] J.M. Goldberg and M.B. Sandler: "Pseudo Natural Pulse Width Modulation for High Accuracy Digital-to-Analogue Conversion," Electronics Letters, Vol. 27, No. 16, 1st August 1991.

[Go92] J.M. Goldberg and M.B. Sandler: "A Digital Signal Processing Algorithm for the Linearization of PWM Based DACs for Digital Audio Applications," Signal Processing VI, EUSIPCO Proceedings, Brussels, 1992.

[Gr87] "Oversampled Sigma-Delta Modulation," IEEE Transactions on Communications, Vol COM-35, No. 5, May 1987.

[Ha91] M.W. Hauser: "Principles of Oversampling A/D Conversion," Journal of the Audio Engineering Society, Vol. 39, No.1/2, January/February 1991.

[Ha92] M.O.J. Hawksford: "Dynamic Model-Based Linearization of Quantized Pulse-Width Modulation for Applications in Digital-to-Analog Conversion and Digital Power Amplifier Systems," Journal of the Audio Engineering Society, Vol. 40, No. 4, April 1992.

[Hi91] R.E. Hiorns, et al: "Design Limitations for Digital Audio Power Amplification," IEE Colloquium for Digital Audio, London, May 1991.

[Hi92a] R.E. Hiorns: private discussion, 1992.

[Hi92b] R.E. Hioms: private discussion, 1992.

[Hi92c] R.E. Hioms: private discussion, 1992.

[Hi92d] R.E. Hioms: "DSP and Circuit Design for PWM D/A Conversion," Ph.D. Thesis, University of London, in preparation, 1992.

[Hi92e] R.E. Hioms: private discussion, 1992.

[Hi92f] R.E. Hioms private discussion, 1992.

[Ho91] U. Horbach and M. Lang: "Design and Implementation of Sigma-Delta D/A Converters with Optimized Loop Filters," Proceedings of IEEE International Symposium on Circuits and Systems, Singapore, 1991.

[Ja84] N.S. Jayant and P. Noll: *Digital Coding of Waveforms,* Prentice-Hall, London, 1984.

[Li78] B. Liu and F. Mintzer: "Calculation of Narrow-Band Spectra By Direct Decimation," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-26, No. 6, December 1978.

[Lo82] B. Lorifeme: *Analog-Digital and Digital-Analog Conversion,* Heyden, London, 1982.

[Ma79] J.H. MacClellan, T.W. Parks, and L.R. Rabiner: "FIR Linear Phase Filter Design Program," in *Programs for Digital Signal Processing,* IEEE Press, New York, 1979.

[Ma70] J.D. Martin: "Theoretical Efficiencies of Class D Power Amplifiers," Proc. IEEE, Vol. 117, No. 6., June 1970.

[Ma91] V.J. Mathews: "Adaptive Polynomial Filters," IEEE Signal Processing Magazine, July 1991.

[Ma87] Y. Matsuya, et al: "A 16-bit oversampling A-to-D conversion technology using triple-integration noise shaping," IEEE Journal of Solid-State Circuits, Vol. 22, December 1987.

[Ma89] Y. Matsuya, et al: "A 17-bit Oversampling D-to-A Conversion Technology Using Multistage Noise Shaping," IEEE Journal of Solid-State Circuits, Vol. 24, No. 4, August 1989.

[Mc79] J.H. McClellan, T.W. Parks, and L.R. Rabiner "FIR Linear Phase Filter Design Program," in *Programs for Digital Signal Processing,* IEEE Press, 1979.

[Me91] P.H. Mellor, et al: "Reduction of Spectral Distortion in Class D Amplifiers by an Enhanced Pulse Width Modulation Sampling Process," IEE Proceedings-G, Vol. 138, No. 4, August 1991.

[Mi60] D. Middleton: *An Introduction to Statistical Communication Theory*, McGraw-Hill, 1960.

[Na87] P.J. Naus, et al: "A CMOS Stereo 16-bit D/A Converter for Digital Audio," IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 3, June 1987.

[Na91a] R. Nawrocki: "Noise Shaping Filter Design," Collection of Computer Programs written at King's College London, September 1991.

[Na91b] R. Nawrocki, J.M. Goldberg, and M.B. Sandler: "Information Theoretic Constraints on Noise Shaping Networks with Minimum Noise Pwer Gain," submitted to IEEE Transactions on Circuits and Systems, December 1991.

[Nu81] A.H. Nutall: "Some Windows with Very Good Sidelobe Behavior," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-29, No. 1, February 1981.

[Ol62] F.W.J. Olver: *Mathematical Tables National Physical Laboratory - Vol. 6: Tables for Bessel Functions of Moderate or Large Orders*, H.M.S.O., London, 1962.

[Op89] A.V. Oppenheim and R.W. Schafer: *Discrete-Time Signal Processing*, Prentice-Hall, London, 1989.

[Os73] A. Ostrowski: *Solution of Equations in Euclidean and Banach Spaces*, Academic Press, New York, 1973.

[Pa65] P.F. Panter: *Modulation, Noise and Spectral Analysis: Applied to Information Transmission*, McGraw-Hill, New York, 1965.

[Pa92] A.C. Paul: Transfer Thesis, University of London, 1992.

[Pa92a] A.C. Paul: "Simulate Version 3.1" set of computer programs, 1992.

[Ph] P. Philips: "A new Digital to Analog Converter for Compact Disc players," Harmon Kardon, Inc., Technical Advertisement.

[Pr88] J.G. Proakis and D.G. Manolakis: *Introduction to Digital Signal Processing*, Macmillan, London, 1988.

[Ro65] H.E. Rowe: *Signals and Noise in Communications Systems*, D. Van Nostrand Company, Inc., London, 1965.

[Ru81] W.J. Rugh: *Nonlinear System Theory The Volterra-Wiener Approach*, Johns Hopkins University Press, Baltimore, 1981.

[Sa83] M.B. Sandler: *Investigation By Simulation of a Digitally Addressed Audio Power Amplifier*, Ph.D. Thesis, University of Essex, October 1983.

[Sa86] M.B. Sandler: "Progress Towards a Digital Power Amplifier," Presented at the 80th Convention of the Audio Engineering Society, Montreux, 1986.

[Sc89] H.R. Schwarz: *Numerical Analysis A Comprehensive Introduction*, Wiley, Chichester, 1989.

[Sp62] H.A. Spang III and P.M. Schultheiss: "Reduction of Quantization Noise By Use of Feedback," IEE Transactions on Communications Systems, Vol. CS-10, December, 1962.

[Te90] G.C. Temes and J.C. Candy: A Tutorial Discussion of the Oversampling Method for A/D and D/A Conversion," Proceedings of IEEE International Symposium on Circuits and Systems, New Orleans, 1990.

[Te78] S.K. Tewksbury and R.W. Hallock: "Oversampled, Linear Predictive, and Noise Shaping Coders of Order $N>1$," IEEE Transactions on Circuits and Systems, Vol. CAS-25, July 1978.

[Va87] P.P. Vaidyanathan: "Design and Implementation of Digital FIR Filters," in *Handbook of Digital Signal Processing Engineering Applications*, (ed. D.F. Elliot), Academic Press, Inc., London, 1987.

[Va82a] R.J. Van de Plassche and E.J. Dijkmans: A Monolithic 16-Bit D/A Conversion System for Digital Audio *Digital Audio Collected Papers From AES Premiere Conference*, New York, 1982.

[Va82b] R.J. Van de Plassche: "Dynamic element matching puts trimless converters on chip," Electronics, Vol. 56, No. 12, 16 June 1982.

[Va84] J. Vanderkooy and S. Lipshitz: "Resolution Below the Least Significant Bit in Digital Systems with Dither," Journal of the Audio Engineering Society, Vol. 32, March 1984.

[Va89] J. Vanderkooy and S. Lipshitz: "Digital Dither: Signal Processing with Resolution Far Below the Least Significant Bit," Proceedings of the Audio Engineering Society 7th International Conference, Toronto, May 1989.